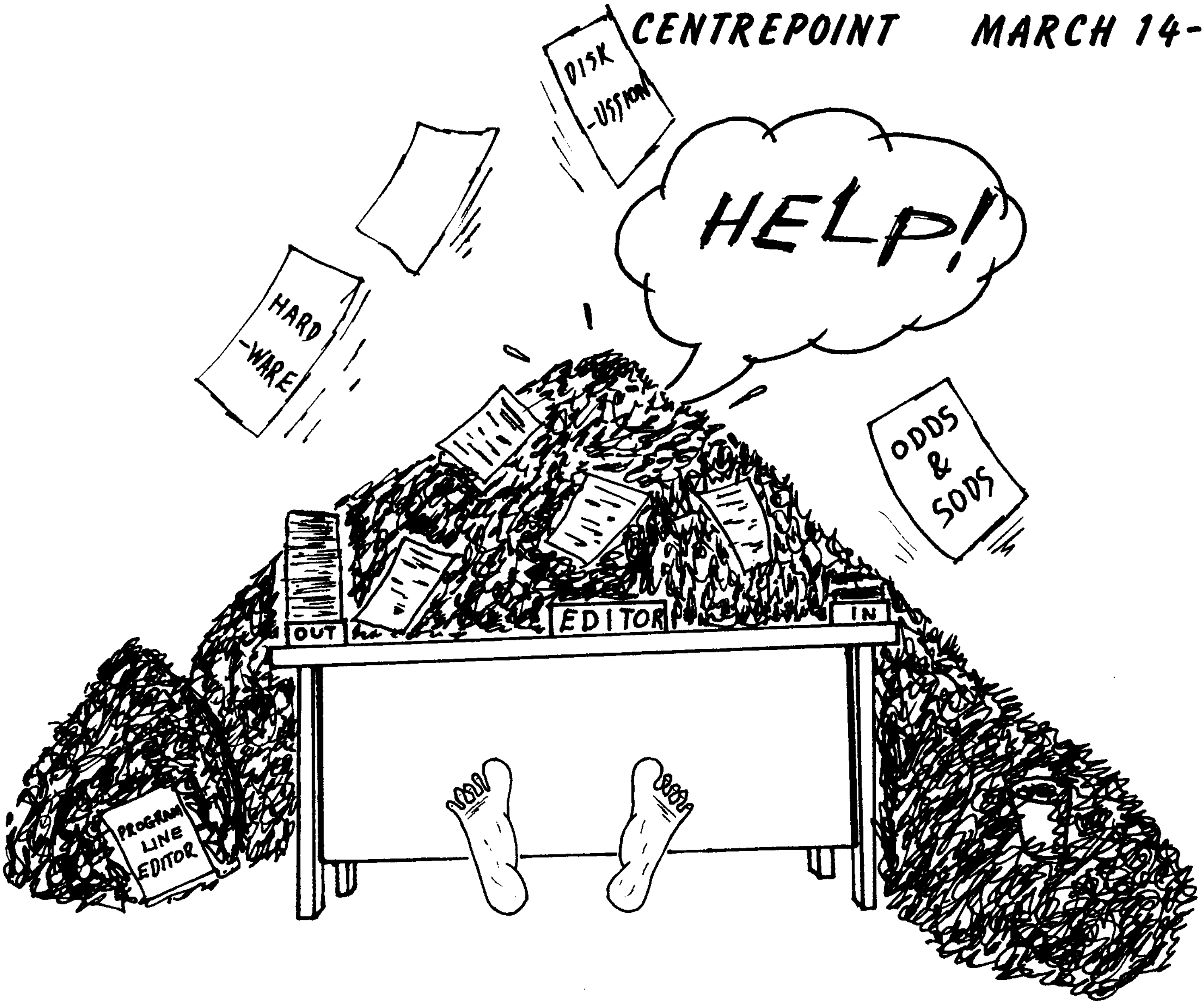
ATARI COMPUTER ENTHUSIASTS [N.S.W.]

INSIDE INFO No. 11 February 1984

AUSTRALIAN
PERSONAL COMPUTER SHOW
CENTREPOINT MARCH 14-17



REGISTERED BY AUSTRALIA POST PUBLICATION No. NHB 5944

INFORMATION

Atari Computer Enthusiasts (N.S.W.) is an independent, non-profit computer users' group loosely affiliated with Atari Computer Enthusiasts in the U.S.A. We have no connections with ATARI, Inc. or their Australian distributors, Futuretronics Australia Pty. Ltd. Our aims include promotion of the various ATARI Home Computer Systems, instructing both beginners and advanced users in programming techniques, exchanging public domain software, hints, tips and ideas amongst members and generally enjoying ourselves. The Club cannot condone software piracy.

COMMITTEE MEMBERS	Home Phone #
Tony Reeve (President)	(02)452 2974
Barry Williams (Vice President)	(02)452 2229
Tony Dinallo (Secretary/Treasurer)	(02)797 0758
Peter Bamford (Meetings Organizer)	(043)42 2655
Peter Dickeson (Cassette S/W Xchge)	(02) 80 2505
Brian Simmons (Disk S/W Xchge)	(045)76 5331
Chris Fitzgerald (Editor)	(02)637 5447
Roberto Romano (Bulletin Board)	(02)327 4898
Jeff Maddock (Librarian)	(02)568 2990

Address all mail to the relevent committee position:-

C/- Atari Computer Enthusiasts (N.S.W.)
G.P.O. Box 4514,
Sydney,
N.S.W. 2001.

MEMBERSHIP FEES

$30 for the first year and $15 thereafter (or $20 and $10 respectively for students under 18 and still at school). Overseas air-mail subscriptions are $10(Aust) extra per year. Please write to the Secretary for further details.

SOFTWARE EXCHANGE

DISKS: $10 each. CASSETTES: $8 each.

Include an extra $1.50 to cover surface postage within Australia. Contact appropriate software exchange for further details.

Overseas members should write for costs first.

Currently, there are 5 titles available to members, on either disk or tape. These are:

INSIDE INFO Vol. 1
INSIDE INFO Vol. 2 with CARTOONS
GAMES #1
UTILITIES #1 and
fig-FORTH.

For every program you write and have accepted by the Software Exchange, you will receive a bonus coupon. This will entitle you to one disk or tape at half price.

We would be pleased to exchange software with other user's groups.

BLANK MEDIA

DISKS: $28 per box of 10.

CASSETTES: $9 per box of 10.

NOTE: Prices may change without notice.

An extra $1.50 should be included to cover postage within Australia. Contact appropriate Software exchange for details.

SPECIAL INTEREST GROUPS (S.I.G.s)

Home phone numbers of contacts are given below. Phone them or write to the Editor with your suggestions, high scores, reviews, hints or tips.

ADVENTURES: Contact Peter Bamford	(043)42 2655
FORTH: Contact John Mattes	(02) 94 5463
ARCADE GAMES: Contact Ken Shiu	(02)534 2120

BULLETIN BOARD SERVICE

This is not yet up and running. Charges and other details have not yet been decided. Contact Bulletin Board Operator for details.

INSIDE INFO

This is our bi-monthly users' group magazine. Articles, etc. should reach the editor at least one month ahead of the release date. Authors of articles printed in INSIDE INFO are entitled to receive a half price coupon for the software exchange.

BACK ISSUES: $4 each plus air-mailing costs.

EXCHANGE SUBSCRIPTIONS: We are keen to exchange magazines with other Atari Users' Groups. Please write to the Secretary for details.

COPYRIGHT: Unless otherwise stated, articles in INSIDE INFO are not copyrighted. However, if any article is reprinted, acknowledgement of our source should be made. Also, please send TWO copies of your magazine to A.C.E. (N.S.W.).

ADVERTISING: Contact the Vice President for costs and conditions. Members' personal adds are free.

REFERENCE LIBRARY

Some books are brought to meetings, others are available only by arrangement with the librarian. Loans are made only to people researching articles or talks for the club.

DISCOUNTS AVAILABLE TO MEMBERS

You may be asked to present your membership card before being given any discounts!

COMPUTERWAVE [5% -cash only] 325 George St, Sydney. (near Wynyard) Ph. (02) 29 1631.

COMPUTERWAVE [5% -cash only] 436 George St, Sydney. (lower ground floor, Grace Bros.) Ph. (02)231 5791.

THE COMPUTER SPOT [5% credit, 10% cash] Shop C4, M.L.C. Centre, Martin Place. Ph. (02)235 2971.

COVER CARE [25% off computer dust covers] P.O. Box 719, Chatswood, N.S.W. 2067. Ph. (02)498 5631.

MEETING DETAILS

Meetings are held at 6.00 P.M. on the first Monday of the month (or the second Monday if it clashes with a public holiday) in the Amenities Room, 7th Floor, OTC House, 32-36 Martin Place, Sydney.

NOTE: You can't enter the building after 6.30PM and you MUST sign in!

MEETING DATES FOR 1984

*6th February 5th March *2nd April
7th May *4th June 2nd July
*13th August 3rd September *8th October
5th November *3rd December.

* indicates release dates for INSIDE INFO.

EDITORIAL

Welcome to the new look INSIDE INFO. As the new "temporary" editor I got a bit carried away and decided to do a "little" reformatting of the magazine.

I don't pretend to be a professional at the job and I don't have anywhere near the knowledge on the Atari Computer that our former editor, Garry Francis had, but I'll try and cope. This brings me to my next point -HELP!!!! The guy behind the pile of scrap paper (on the cover) is no longer seated, he's flat on his back! Please write for the magazine -it's the only way that I can possibly hope to give you a good magazine with plenty of interesting articles. If you can't write, don't worry -I'll do my best to present your article in readable english. If you don't have the knowledge to write an article, make a suggestion for an one you'd like to see -I'll list it in the magazine and perhaps someone else will write it.

As you read you should keep in mind that the new double column format allows a lot more text on a page -one of these pages is roughly the equivalent of one and a half of the old pages. Future issues will not necessarily be 22 pages but more like 14 or 16 pages and this should save the club some money on printing.

Screen dumps used in this issue are used by permission and are copyrighted by Jennifer Brabson (U.S.A.).

Happy Reading, Chris Fitzgerald.

NOTICES

**The December meeting was our last on the premises of COMPUTERWAVE. Since then the committee has been searching to find another, suitable meeting place. We have been extremely fortunate to get back into OTC House for future meetings (OTC House Ameneties room, 7th Floor, 32-36 Martin Place, Sydney).

There are a few rules for members to keep in mind if we are to continue using the premises.

Building security expects all members to SIGN IN -an A.C.E. member will be in the foyer with a book for you to sign. You will not be allowed into the building after 6.30PM!

Keep the place tidy and clean, and don't wander through the building.

**The Australian Personal Computer Show is on again. Entry is going to cost $4, but we should be getting tickets for members which will allow a $1 discount. These tickets will be available at the meeting on the 5th March.

We'll also need plenty of willing volunteers to man our users' group stand during the show -please offer your help, contact the Vice President for further details.

**Due to the Atari 800XL having been accepted for use in New South Wales schools, the Committee has decided to extend the closing date for the Educational Software Competition by TWO months. The new closing date is 2nd April (April meeting). This should help us build up educational software which we can offer to the schools. Get cracking!

**Ken Shiu has opted to organize an Arcade Special Interest Group. All he asks is that interested members keep running high scores of the games they play. You can contact Ken by phoning (02)534-2120 with reviews, hints, tips or suggestions.

**Jeff Maddock recently purchased a SENDATA 700 acoustic coupler/modem. He is interested in contacting other "on-line" members so you can put your modems and software though their paces. If you like, you can leave your name and phone number with him so that a list of on-line users can be printed (with your permission) in a future issue of INSIDE INFO. Contact Jeff by phoning (02)568-2990.

**Some members have returned some of the blank disks which they purchased from the software exchange. Out of three of the returned "faulty" disks we had no trouble formatting two of them.

The club will gladly replace any faulty disks, but please try to format the disk on another disk drive (if possible) if you have any trouble.

**Those of you who missed our users' group Chritmas party missed out on a GREAT night. All up, about 30 people showed up to enjoy each other's company and to generally get to know one another.

Hopefully we'll see you there for Christmas 1984!!!!!

DISKussion

by Ken Scalley. (Winston Hills, N.S.W.)

This is the first of what will, hopefully, be a fairly regular column in which I'll try to share some of my disk experience with you.

Soon after I purchased my 810 disk drive, I found that I was not able to access some of my disk files; the Volume Table of Contents (VTOC) had been damaged. I went to various shops to find out how to fix it and was told- "Format a new disk, then copy the affected files onto it one at a time". I found their answer to be far from satisfactory when it took half an hour to copy all the files on the disk. There had to be a better way!

I began to study up on the Disk Operating System (DOS) and after six months, having read several books and examined the source listings for DOS and the Operating System (OS), I finally developed a program which would fix VTOC (and a lot more!!).

GENERAL DISK LAYOUT

We will start with the overall layout of the standard DOS disk, that is, one where you put a disk into the standard 810 and format it using DOS Option "I". This disk will consist of 40 concentric tracks, each track having 18 sectors and each sector containing 128 bytes. This means that each DOS II formatted disk has a capacity for 720 sectors or 92 160 bytes.

THE BOOT SECTORS

Not all of the 720 sectors are available for disk file storage. DOS uses the first 3 sectors of each disk as BOOT sectors. These sectors enable the system to "boot" (load) DOS when the computer is powered up.

VOLUME TABLE OF CONTENTS

DOS also uses sector 360 (decimal) for VTOC. Sorry, I only have ten fingers and can't get the hang of this hexadecimal (hex) stuff, so all numbers are given in decimal. VTOC is the "index" that DOS uses to keep track of which sectors are in use and which are free.

The layout of the VTOC sector is as follows:

Byte 0: if set to 2, tells DOS that the disk is DOS II format.

Bytes 1 and 2: give the maximum possible number of sectors which can be made available in low byte, hi byte form. This is 707 for DOS II.

Bytes 3 and 4: give the number of currently available sectors, again in low byte, hi byte order. This is the number given in the "FREE SECTORS" message in a disk directory.

Bytes 10 to 99: give the actual sector allocation information. Each byte when expressed as a binary number (i.e. 0's and 1's) consists of 8 bits. The result is that these 90 bytes give 720 (90*8) values, a 1 indicating that the relevent sector is free, or a 0 indicating that the relevent sector is in use.

Due to an oversight in DOS, sector 720 (the last sector on the disk) cannot be addressed even though the 810 is physically able to access it. Hence, DOS also marks this sector as being in use.
For similar reasons, sector 0 is inaccessible; whilst DOS II recognizes sector 0, the 810 does not!

THE DISK DIRECTORY

DOS also uses 8 sectors (361 to 368) for the disk directory. Each sector holds information on 8 disk files, hence the maximum number of files in a directory is 64. Each file entry consists of 16 bytes, in the following format:

Byte 0: this is the flag byte and indicates if the file has been deleted or is "active" and whether it is locked or unlocked.

Bytes 1 and 2: indicate the number of sectors used by the file.

Bytes 3 and 4: indicates the first (starting) sector of the file.

Bytes 5 to 15: are the 11 bytes which contain the filename and its extension.

After formatting your disk, you end up with only 707 of the 720 sectors to use for your programs and that is not the end of it!

GENERAL SECTOR FORMAT

Each of the remaining 707 sectors on the disk contains space for 128 bytes of data. However, DOS II also makes use of the last 3 bytes of these sectors. These bytes contain:

1) the file number (0 to 63): indicating to DOS which file the sector belongs to.

2) a pointer: which tells DOS where the next sector in the file is located. The last sector in a file points to 0 as the next sector.

3) the number of bytes in use in that sector.

That's it for this issue. Hopefully, if there aren't too many complaints from readers, the editor will let me back next issue. If he does, the next article will look at the above in greater detail and hopefully, we will come up with a simple BASIC program which will allow you to look at any sector on a disk, without using DOS.

[I'm sure he will -ed.]

Graphs

by Allan R. Smyth. (Hurstville, N.S.W.)

CURVES

This program sketches up to 3 curves on the screen at one time. Screen 1 asks you how many curves. Press 1, 2 or 3 followed by RETURN.

Screen 2 asks you to enter the function starting at the flashing cursor. This must be the correct syntax for BASIC, e.g. y=3*x+7 will be accepted, but y=3x+7 will generate a function error. Use x as the independent variable. You have the option of maintaining the current function shown on the bottom of the screen by pressing RETURN.

Screens 3 and 4 are the same as screen 2, but for a second and third function (if required).

Screen 5 asks whether you require axes. Unless you are plotting a part of a function that does not cross any axes, press Y then RETURN.

Screen 6 asks whether to use degrees or radians. When trigonometric functions are involved, press D or R as appropriate followed by RETURN.

Screens 7 and 8 are used to enter the domain of the function(s). Enter the minimum value of x then the maximum value of x.

Screen 9 of the program has 2 options.

1. The program will work out the MAX/MIN of the curves and then scale the screen appropriately. This is very slow and is inappropriate for curves which have asymptotes. To use this option, press RETURN without a number.
2. You may enter a number at MAX/MIN and the curve will be drawn for the range -MAX<y<MAX. After the curves are drawn, press SELECT to continue and you will be given the chance to redraw the curves under a different domain and range.

NEWTON'S METHOD

This program uses Newton's method to find the zeroes of curves. For screen 1, enter the function for which you require the zero to be found. The program will check for correct syntax. For screen 2, enter your first estimate of the zero. This can usually be way off the mark as the program converges quickly.

You will see each new approximation flashed on the screen until two consecutive answers are the same. You are then given the option of changing the function by pressing RETURN without any digit or of entering a new estimate for other zeroes.

There is one problem with this program. Some functions have consecutive estimates which do not converge, but oscillate. Although rare, this situation can be overcome by pressing BREAK to stop the program.

FUNCTION NOTATION

This program is useful for year 7 to demonstrate substitution and function notation. Screen 1 asks for a function of the form y=f(x).

Screen 2 asks for an x value. You may enter either a numeral or pronumeral. The program runs in an infinite loop. Press BREAK to stop the program.

WALLPAPER

This program illustrates 4-fold symmetry. There are 4 axes of symmetry, so the patterns are repeated 8 times. Because the patterns are random, some look better than others, but it does remind you of wallpaper. This program was modified from an APPLE program called kaleidoscope.

SNOWFLAKE

This is a demonstration of rotational symmetry.

THE GRAPHICS MACHINE

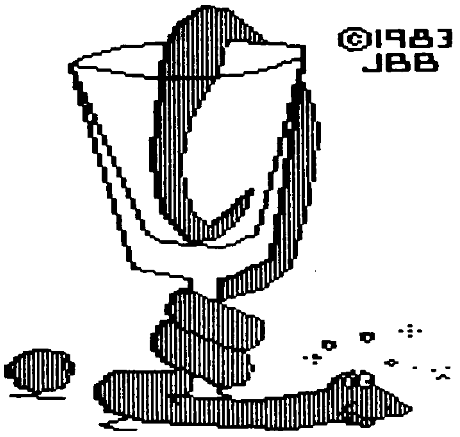
Here is a series of programs which demonstrate some of the work done in class. It was actually my first program for the ATARI and was written for a parent-teacher night. The demos include:

1. Envelopes with straight lines - cardoid, astroid, nephroid and mystic circles.
2. Polar graphs - some unusual ones.
3. Envelopes with circles - limacon with loop, trefoil.

```
1 REM ################################
2 REM #          CURVES              #
3 REM #     by Allan R. Smyth        #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1984         #
7 REM ################################
10 OPEN #1,4,0,"K:"
20 GRAPHICS 0:SETCOLOR 2,0,0:? "THIS P
ROGRAM WILL PLOT UP TO 3":? "CURVES ON
 THE SCREEN AT ONCE"
30 TRAP 20:? :? "HOW MANY CURVES";:INP
UT CURVES
40 IF CURVES>3 OR CURVES<1 OR CURVES<>
INT(CURVES) THEN 20
50 OPEN #2,4,0,"E:"
60 FOR ZX=1 TO CURVES:SETCOLOR 2,0,0:?
 CHR$(125)
70 ? "INPUT FUNCTION ";ZX:POSITION 2,1
9:? "PRESS RETURN TO MAINTAIN CURRENT
":? "FUNCTION. CURRENT FUNCTION IS..."
80 POSITION 2,21:LIST 1000*ZX:POSITION
 2,22:FOR I=1 TO 4:? CHR$(254);:NEXT I
90 TRAP 760
100 POSITION 2,8:? "Y=";:POKE 764,255
110 X= NOT X:POKE 752,X:? CHR$(254);:F
OR Y=1 TO 16:IF PEEK(764)=255 THEN NEX
T Y:GOTO 110
120 POP :POKE 752,0:Z=0
130 GET #2,Y:IF Y<>155 THEN Z=Z+1:GOTO
 130
140 IF Z=0 THEN 190
150 POKE 559,0
160 POSITION 2,8:FOR I=1 TO 4:? CHR$(2
55);:NEXT I:? 1000*ZX:POSITION 2,15:?
"CONT"
170 POSITION 2,3:POKE 842,13:STOP
180 POKE 842,12:X=0.93:GOSUB 1000*ZX:?
 CHR$(125):POKE 559,34
190 NEXT ZX
200 ? CHR$(125);"DO YOU REQUIRE AXES (
Y/N)"
210 GET #1,KEY1:IF KEY1<>89 AND KEY1<>
78 THEN 210
220 ? CHR$(125);"DEGREES OR RADIANS (D
/R)"
230 GET #1,KEY:IF KEY=68 THEN DEG
240 IF KEY=82 THEN POKE 251,0
250 IF KEY<>68 AND KEY<>82 THEN 220
260 TRAP 260:? CHR$(125);"FOR THE DOMA
IN PLEASE INDICATE":? :? "MINIMUM VALU
```

```
E OF X";:INPUT MINX
270 TRAP 270:POSITION 2,5:? "MAXIMUM V
ALUE OF X";:INPUT MAXX
280 TRAP 310:? CHR$(125);"IF YOU WISH
TO RESTRICT THE RANGE":? :? "PLEASE EN
TER A VALUE FOR MAX/MIN"
290 ? :? "OTHERWISE PRESS RETURN AND P
ROGRAM":? :? "WILL SCALE CURVE TO FILL
 ENTIRE SCREEN":? :? "INPUT MAX/MIN";
300 INPUT MAXY1:MAXY=MAXY1:MINY=-MAXY1
:GOTO 400
310 MAXY1=2000:MAXY=-2000:MINY=2000
320 GRAPHICS 18:? #6;"WORKING...":FOR
Z=1 TO CURVES
330 FOR ZX=0 TO 159:SOUND 0,RND(0)*200
,10,6
340 X=(MAXX-MINX)/159*ZX+MINX
350 ON Z GOSUB 1000,2000,3000
360 IF Y>MAXY THEN MAXY=Y
370 IF Y<MINY THEN MINY=Y
380 NEXT ZX
390 NEXT Z
400 SOUND 0,0,0,0
410 IF MINY>0 AND KEY1=89 THEN MINY=0
420 IF MAXY<0 AND KEY1=89 THEN MAXY=0
430 IF MAXY>MAXY1 THEN MAXY=MAXY1
440 IF MINY<-MAXY1 THEN MINY=-MAXY1
450 ZX=-159*MINX/(MAXX-MINX):ZY=-79*MI
NY/(MAXY-MINY):GRAPHICS 7:SETCOLOR 1,6
,8:SETCOLOR 2,0,14
460 IF CURVES<3 THEN SETCOLOR 2,0,0
470 COLOR 1:IF CURVES=1 THEN COLOR 2
480 IF MINX<=0 AND MAXX>=0 THEN PLOT Z
X,0:DRAWTO ZX,79
490 COLOR 1:IF CURVES=1 THEN COLOR 2
500 IF MINY<=0 AND MAXY>=0 THEN PLOT 0
,79-ZY:DRAWTO 159,79-ZY
510 FOR Z=1 TO CURVES
520 ZY=1
530 LIST 1000*Z:? CHR$(28);:FOR I=1 TO
 4:? CHR$(254);:NEXT I
540 COLOR Z:FOR ZX=0 TO 159
550 X=(MAXX-MINX)/159*ZX+MINX
560 TRAP 620
570 ON Z GOSUB 1000,2000,3000
580 Y=(Y-MAXY)*79/(MINY-MAXY)
590 IF ZY=0 THEN DRAWTO ZX,Y
600 IF ZY=1 THEN PLOT ZX,Y
610 ZY=0:GOTO 630
620 ZY=1
630 NEXT ZX
640 NEXT Z
650 ? CHR$(125);"DOMAIN ";MINX;"<=X<="
;MAXX:? "RANGE ";MINY;"<=Y<=";MAXY:POK
E 752,1
660 ? "PRESS SELECT TO CONTINUE":? "PR
ESS OPTION TO END PROGRAM";
670 ZX=PEEK(53279):IF ZX=3 THEN END
680 IF ZX<>5 THEN 670
690 ? CHR$(125);"PRESS START TO RESTAR
T PROGRAM"
700 ? "PRESS OPTION TO END PROGRAM";
710 FOR ZX=1 TO 500:NEXT ZX
720 ZX=PEEK(53279):IF ZX=3 THEN END
730 IF ZX=5 THEN GRAPHICS 0:SETCOLOR 2
,0,0:GOTO 260
740 IF ZX=6 THEN RUN
750 GOTO 720
760 POKE 559,34:IF ERR=8 THEN 190
770 ? CHR$(125);"FUNCTION ERROR";CHR$(
253):GOTO 70
1000 Y=COS(2*X)
1010 RETURN
2000 Y=SIN(X)
2010 RETURN
3000 Y=SIN(X)+COS(2*X)
3010 RETURN
```

```
1 REM ################################
2 REM #            NEWTON            #
3 REM #      by Allan R. Smyth       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #         February 1984        #
7 REM ################################
10 OPEN #2,4,0,"K:"
20 OPEN #1,4,0,"E:"
30 ? CHR$(125)
40 GOTO 260
50 ? "INPUT YOUR ESTIMATE, PLEASE":INP
UT X2
60 GRAPHICS 2
70 SETCOLOR 2,0,0
80 ? #6;CHR$(125):? #6;X2:GOSUB 160
90 IF X2<>X1 THEN X2=X1:GOTO 80
100 LIST 1000
110 ? CHR$(28);:FOR I=1 TO 5:? CHR$(25
4);:NEXT I:?
120 ? "PRESS RETURN TO CHANGE FUNCTION
,":? "ANY OTHER KEY TO CONTINUE";
130 GET #2,X
140 IF X=155 THEN GRAPHICS 0:SETCOLOR
2,0,0:GOTO 260
150 GRAPHICS 0:SETCOLOR 2,0,0:GOTO 50
160 TRAP 310:X=X2:GOSUB 1000:Y0=Y:X=X2
+1E-06:GOSUB 1000:Y1=Y:X=X2-1E-06:GOSU
B 1000:Y2=Y:DIFF=(Y1-Y2)*500000
170 X1=X2-Y0/DIFF:RETURN
180 X= NOT X:POKE 752,X:? CHR$(254);:F
OR Y=1 TO 16:IF PEEK(764)=255 THEN NEX
T Y:GOTO 180
190 POKE 752,0:TRAP 290
200 GET #1,Y:IF Y<>155 THEN 200
210 POKE 559,0
220 POSITION 2,8:FOR I=1 TO 5:? CHR$(2
55);:NEXT I:? 1000:? "CONT":? "CONT":P
OSITION 2,15:? "CONT"
230 POSITION 2,3
240 POKE 842,13:STOP
250 POKE 842,12:? CHR$(125):POKE 559,3
4:X=0.93:TRAP 290:GOSUB 1000:RETURN
260 SETCOLOR 2,0,0
270 ? "INPUT FUNCTION":POSITION 2,8:?
"Y=";:POKE 764,255:GOSUB 180
280 ? CHR$(125):GOTO 50
290 POKE 559,34
300 ? CHR$(125);"FUNCTION ERROR";CHR$(
253):GOTO 260
310 GRAPHICS 0:SETCOLOR 2,0,0:? "DERIV
ATIVE UNDEFINED AT THIS POINT":GOTO 50
1000 Y=SIN(X)
1010 RETURN
```



```
1 REM ################################
2 REM #          FUNCTION            #
3 REM #      by Allan R. Smyth       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1984         #
7 REM ################################
10 DIM A$(20),F$(25),X$(10)
20 GRAPHICS 0:SETCOLOR 2,0,0
30 OPEN #1,4,0,"E:"
40 SETCOLOR 2,0,0
50 ? "INPUT FUNCTION":POSITION 2,8:? "
Y=";:POKE 764,255
60 GOSUB 320
70 GOSUB 410
80 GRAPHICS 2:SETCOLOR 2,0,0
90 ? #6;A$:? #6;"X=?"
100 ? "INPUT X VALUE";:INPUT X$
110 Z=0
120 TRAP 460:X=VAL(X$):IF LEN(STR$(X))
<LEN(X$) THEN GOTO 460
130 IF Z<>1 THEN GOSUB 530
140 ? #6;CHR$(125)
150 POSITION 0,0
160 ? #6;A$
170 ? #6;"when X=";X$
180 FOR X=1 TO LEN(A$):IF A$(X,X)="X"
THEN ? #6;X$;:GOTO 200
190 ? #6;A$(X,X);
200 NEXT X:? #6
210 IF Z<>1 THEN ? #6;"Y=";Y:GOTO 230
220 ? #6
230 ? #6,"or":? #6:? #6;F$
240 ? #6;F$(1,1);:IF Z=1 THEN ? #6;X$;
:GOTO 260
250 ? #6;"(";X$;")";
260 FOR X=5 TO LEN(F$):IF F$(X,X)="X"
THEN ? #6;X$;:GOTO 280
270 ? #6;F$(X,X);
280 NEXT X:? #6
290 IF Z<>1 THEN ? #6;"    =";Y:GOTO 3
10
300 ? #6
310 ? "To change function press RETURN
":? :? :GOTO 100
320 X= NOT X:POKE 752,X:? CHR$(254);:F
OR Y=1 TO 16:IF PEEK(764)=255 THEN NEX
T Y:GOTO 320
330 POP :POKE 752,0
340 TRAP 510
350 GET #1,Y:IF Y<>155 THEN 350
360 POKE 559,0
370 POSITION 2,8:FOR I=1 TO 6:? CHR$(2
55);:NEXT I:? 530:? "CONT":? "CONT":?
"CONT":? "CONT":? "CONT"
380 POSITION 2,3
390 POKE 842,13:STOP
400 POKE 842,12:X=9.3:TRAP 510:GOSUB 5
30:? CHR$(125):RETURN
410 ? CHR$(125):LIST 530:? CHR$(28);:F
OR I=1 TO 6:? CHR$(31);:NEXT I:X=3
420 IF PEEK(93)=0 THEN 450
430 A$(X,X)=CHR$(PEEK(93)+32):? CHR$(3
1);:X=X+1:GOTO 420
440 POKE 559,34
450 A$(1,2)="Y=":F$(1,5)="F(X)=":F$(6)
=A$(3):? CHR$(125):RETURN
460 IF PEEK(195)=11 THEN ? "DIVISION B
Y ZERO IS NOT POSSIBLE";CHR$(253):? :?
 :GOTO 100
470 TRAP 510:Z=1:FOR X=LEN(X$)+1 TO 2
STEP -1
480 X$(X,X)=X$(X-1,X-1)
490 NEXT X
500 X$(1,1)="(":X$(LEN(X$)+1,LEN(X$)+1
)=")":GOTO 130
510 POKE 559,34:IF PEEK(195)=5 THEN RU
N
520 ? CHR$(125);"FUNCTION ERROR";CHR$(
253):GOTO 50
530 Y=COS(X)
540 RETURN
```

```
1 REM ################################
2 REM #          WALLPAPER           #
3 REM #      by Allan R. Smyth       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1984         #
7 REM ################################
10 GRAPHICS 2:? #6;"WALLPAPER":FOR R=0
 TO 1500:NEXT R
20 T=2
30 POKE 77,0
40 FOR I=20 TO 2000 STEP 10
50 GRAPHICS 23:CO=0
60 FOR J=0 TO T
70 FOR K=19 TO 0 STEP -1
80 COLOR CO
90 R=39*RND(0)
100 C=R+I+J+K:C=C-33*INT(C/33)
110 PLOT R,C:PLOT 39-R,C:PLOT R,39-C:P
LOT 39-R,39-C
120 PLOT R,40+C:PLOT 39-R,40+C:PLOT R,
79-C:PLOT 39-R,79-C
130 PLOT C,R:PLOT 39-C,R:PLOT C,39-R:P
LOT 39-C,39-R
140 PLOT C,40+R:PLOT 39-C,40+R:PLOT C,
79-R:PLOT 39-C,79-R
150 PLOT 40+C,R:PLOT 79-C,R:PLOT 40+C,
39-R:PLOT 79-C,39-R
160 PLOT 80+C,R:PLOT 119-C,R:PLOT 80+C
,39-R:PLOT 119-C,39-R
170 PLOT 120+C,R:PLOT 159-C,R:PLOT 120
+C,39-R:PLOT 159-C,39-R
180 PLOT 40+C,40+R:PLOT 79-C,40+R:PLOT
 40+C,79-R:PLOT 79-C,79-R
190 PLOT 80+C,40+R:PLOT 119-C,40+R:PLO
T 80+C,79-R:PLOT 119-C,79-R
200 PLOT 120+C,40+R:PLOT 159-C,40+R:PL
OT 120+C,79-R:PLOT 159-C,79-R
210 PLOT 40+R,C:PLOT 79-R,C:PLOT 40+R,
39-C:PLOT 79-R,39-C
220 PLOT 80+R,C:PLOT 119-R,C:PLOT 80+R
,39-C:PLOT 119-R,39-C
230 PLOT 120+R,C:PLOT 159-R,C:PLOT 120
+R,39-C:PLOT 159-R,39-C
240 PLOT 40+R,40+C:PLOT 79-R,40+C:PLOT
 40+R,79-C:PLOT 79-R,79-C
250 PLOT 80+R,40+C:PLOT 119-R,40+C:PLO
T 80+R,79-C:PLOT 119-R,79-C
260 PLOT 120+R,40+C:PLOT 159-R,40+C:PL
OT 120+R,79-C:PLOT 159-R,79-C
270 CO=CO+1:CO=CO-4*INT(CO/4)
280 NEXT K:NEXT J:NEXT I
290 END
```

```
1 REM ################################
2 REM #          SNOWFLAKES          #
3 REM #      by Allan R. Smyth       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1984         #
7 REM ################################
9 GRAPHICS 2:POSITION 5,4:? #6;"SNOWfl
akes":FOR X=0 TO 1000:NEXT X
10 GRAPHICS 2:? #6;"ORDER OF SYMMETRY"
:INPUT N
20 ? #6;"how many points":INPUT P
30 DEG :A=360/N:C=1
35 GRAPHICS 23
40 FOR Y=1 TO P
50 TH=A*RND(0)
60 R=75*RND(0)-30
70 COLOR C
80 FOR X=0 TO 360 STEP A
90 PLOT R*COS(X+TH)+80,R*SIN(X+TH)+48
100 PLOT R*COS(X-TH)+80,R*SIN(X-TH)+48
110 NEXT X
120 C=C+1:IF C>3 THEN C=1
130 NEXT Y
140 GOTO 35
```

```
1 REM ################################
2 REM #    THE GRAPHICS MACHINE      #
3 REM #      by Allan R. Smyth       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1984         #
7 REM ################################
10 DEG :POKE 77,0:GRAPHICS 18:SETCOLOR
 0,1,6:SETCOLOR 1,6,6:SETCOLOR 2,12,8:
SETCOLOR 4,4,10
20 POSITION 8,3:? #6;"THE":POSITION 6,
5:? #6;"GRAPHICS":POSITION 6,7:? #6;"m
achine":FOR T=1 TO 3000:NEXT T
30 FOR Z=2 TO 5:GRAPHICS 18
40 IF Z=2 THEN POSITION 6,5:? #6;"CARD
IOID"
50 IF Z=3 THEN POSITION 6,5:? #6;"neph
roid"
60 IF Z=4 THEN POSITION 7,5:? #6;"tref
oil"
70 IF Z=5 THEN POSITION 3,5:? #6;"4 LE
AF CLOVER"
80 FOR T=1 TO 2000:NEXT T:GRAPHICS 24:
COLOR 1:SETCOLOR 1,12,8:SETCOLOR 2,7+Z
,12:SETCOLOR 4,2*Z,6
90 FOR X=0 TO 360 STEP 4
100 A=90*COS(X)+160:B=90*SIN(X)+91
110 Y=Z*X
120 C=90*COS(Y)+160:D=90*SIN(Y)+91
130 PLOT A,B
140 DRAWTO C,D
150 NEXT X
160 FOR T=1 TO 2000:NEXT T
170 NEXT Z
180 GRAPHICS 18:POSITION 6,5:? #6;"AST
ROID":FOR T=1 TO 2000:NEXT T
190 GRAPHICS 24:COLOR 1:SETCOLOR 2,4,1
4:SETCOLOR 4,10,10
200 Y1=95:Y2=95
210 FOR X=65 TO 160 STEP 5
220 PLOT 160,Y1:DRAWTO X,95:DRAWTO 160
,Y2
230 Y1=Y1-5:Y2=Y2+5
240 NEXT X
250 Y1=95:Y2=95
260 FOR X=255 TO 160 STEP -5
270 PLOT 160,Y1:DRAWTO X,95:DRAWTO 160
,Y2
280 Y1=Y1-5:Y2=Y2+5
290 NEXT X
300 FOR T=1 TO 2000:NEXT T
310 GRAPHICS 18:SETCOLOR 0,11,6:SETCOL
OR 2,3,6:SETCOLOR 4,9,10:POSITION 3,5:
? #6;"MYSTIC CIRCLES"
320 FOR T=1 TO 2000:NEXT T
330 GRAPHICS 24:COLOR 1:SETCOLOR 1,12,
4:SETCOLOR 2,15,14:SETCOLOR 4,4,6
340 FOR X=0 TO 340 STEP 20
350 A=90*COS(X)+160:B=90*SIN(X)+91
360 FOR Y=X TO 340 STEP 20
370 C=90*COS(Y)+160:D=90*SIN(Y)+91
380 PLOT A,B
390 DRAWTO C,D
400 NEXT Y
410 NEXT X
420 FOR T=1 TO 2000:NEXT T
430 GRAPHICS 18:POSITION 4,5:? #6;"POL
AR graphs":FOR T=1 TO 2000:NEXT T
440 FOR B=1 TO 5
450 GRAPHICS 24:SETCOLOR 1,4,2:SETCOLO
R 2,1+B,10:SETCOLOR 4,14-B,6:COLOR 1:A
=0:GOSUB 530
460 PLOT X,Y
470 FOR A=3 TO 360 STEP 3
480 GOSUB 530:DRAWTO X,Y
490 NEXT A
500 FOR T=1 TO 2000:NEXT T
510 NEXT B
520 GOTO 600
530 ON B GOSUB 550,560,570,580,590
540 X=R*COS(A)*90+160:Y=R*SIN(A)*90+96
:RETURN
550 R=COS(4*57.3*(SIN(2*A))):RETURN
560 R=COS(2*57.3*SIN(2*A)):RETURN
570 R=(COS(20*A)+5)/6:RETURN
580 R=COS(4*A):RETURN
590 R=SIN(2*57.3*COS(4*A)):RETURN
600 POKE 77,0:GRAPHICS 18:POSITION 2,3
:? #6;"envelopes using":POSITION 6,4:?
 #6;"circles"
610 POSITION 0,6:? #6;"1. LIMACON WITH
 LOOP"
620 POSITION 0,8:? #6;"2. TREFOIL":FOR
 T=1 TO 3000:NEXT T
630 A=2
640 GRAPHICS 24:SETCOLOR 1,8+A,6:SETCO
LOR 2,A,10:SETCOLOR 4,3+A,4:COLOR 1
650 FOR TH=0 TO 360 STEP 10
660 X=A*10*COS(TH)+160:Y=10*A*SIN(TH)+
96
670 IF A=2 THEN R=SQR((X-120)^2+(Y-96)
^2)
680 IF A=5 THEN PH=TH-120*INT(TH/120):
PH=60-ABS(PH-60):R=50*SIN(PH)
690 PH=0:GOSUB 760:PLOT X1,Y1
700 FOR PH=10 TO 360 STEP 10
710 GOSUB 760:DRAWTO X1,Y1
720 NEXT PH
730 NEXT TH
740 IF A=5 THEN GOTO 770
750 A=5:GOTO 640
760 X1=R*COS(PH)+X:Y1=R*SIN(PH)+Y:RETU
RN
770 GRAPHICS 18:POSITION 7,5:? #6;"HEl
p!/":FOR T=1 TO 2000:NEXT T
780 GRAPHICS 19
790 C=1
800 A=19:B=20
810 FOR X=0 TO A
820 COLOR C
830 FOR Y=X TO B
840 IF B=X THEN B=19-X:POP :GOTO 900
850 IF A=10 THEN PLOT 10,10:PLOT 9,10:
PLOT 9,9:PLOT 10,9:POP :POP :GOTO 1050
860 PLOT X,Y
870 DRAWTO A,Y
880 B=B-1
890 NEXT Y
900 A=A-1
910 C=C+1
920 IF C=4 THEN C=1
930 NEXT X
940 GOTO 1050
950 X=0:FOR T=1 TO 30:NEXT T
960 A=3:B=8:C=12:D=10:E=9:F=4:G=0:H=0
970 SETCOLOR 0,C,D:SETCOLOR 1,E,F:SETC
OLOR 2,A,B
980 FOR T=1 TO 20:NEXT T
990 SETCOLOR 0,E,F:SETCOLOR 1,A,B:SETC
OLOR 2,C,D
1000 FOR T=1 TO 20:NEXT T
1010 SETCOLOR 0,A,B:SETCOLOR 1,C,D:SET
COLOR 2,E,F
1020 FOR T=1 TO 20:NEXT T
1030 X=X+1:IF X<100 THEN 970
1040 GOTO 10
1050 C=3
1060 A=39:B=20
1070 FOR X=20 TO A
1080 COLOR C
1090 FOR Y=X-20 TO B
1100 IF B=X-20 THEN B=39-X:POP :GOTO 1
160
1110 IF A=30 THEN PLOT 29,9:PLOT 29,10
:PLOT 30,10:PLOT 30,9:POP :POP :GOTO 9
50
1120 PLOT X,Y
1130 DRAWTO A,Y
1140 B=B-1
1150 NEXT Y
1160 A=A-1
1170 C=C-1
1180 IF C=0 THEN C=3
1190 NEXT X
1200 GOTO 950
```

ODDS AND SODS

Tip #1: Young Christopher Scalley showed me this one. When you boot your disk drive and the message "BOOT ERROR" appears on the screen you can stop the drive by hitting "CTRL 1". After the normal cut-out time, when your drive stops spinning, you can insert the correct disk, hit "CTRL 1" again and the disk will boot -with the minimum of fuss!

*****

Tip #2: Chris' dad, Ken Scalley showed me a way of loading a BINARY LOAD file from BASIC. Of course, the file would have to be one that will RUN with the BASIC cartridge in place (i.e. a compiled BASIC program). While in BASIC you should type:

```
     OPEN #1,4,0,"D:filename.OBJ"
     X=USR(5576)
```

*****

Tip #3: Lately, a number of magazines have advised that there is a way out of the infamous Atari keyboard lock-up, apart from this, I received a letter from FUTURETRONICS which mentioned the same method. I should mention that I have had absolutely NO luck with it but perhaps it will work in some situations. Let us know if you get it to work. (Hope you've got enough fingers!)

Hold down the START, SELECT and OPTION keys simultaneously. While holding these down press and hold down the following keys in this order: SHIFT, CTRL, P, M and %. Then LIST the program and edit.

*****

Tip #4: A lot of people don't realise that they don't need to have both DOS files on their disks. DOS SYS is the only one necessary to allow a disk to "boot". DUP SYS is the Disk Utilities Package and is only used when you want to enter the DOS MENU (i.e. when you type "DOS" from BASIC. Generally an AUTO-BOOT disk, such as one with the AtariWriter printer driver on it, would not require DUP SYS -freeing another 42 for ready use.

To get only the one DOS file on a disk you must write both, DOS SYS and DUP SYS using DOS option "H", and then delete DUP SYS using DOS option "D".

*****

Recently, I wrote to FUTURETRONICS for a solution to the problem of the computer and/or its peripherals "going to sleep". They advised that this could be rectified by substituting a Revision B ROM into the Operating System. They also said to forward $7.43 and they would send it to me. Unfortunately, when it arrived it had no instructions to tell me which chip it replaced on the ROM board. To top it off, when I finally worked out where it went, I found that the chip I received was the original Version A [PEEK(58383) will equal 249 -not 56 like in the States] -ARRGH!!!, more time wasted!!!

The hassle with the system "going to sleep" is that when I get a print-out, the OS times out, my printer skips a line and then reprints half a sentence: and that's very annoying when when your trying to print a magazine!

*****

ATARI's 410 (and now the 1010) program recorder has always been famous (infamous?) for being incredibly expensive for what it is. It's strange, but the ATARI units are three times the price of the COMMODORE cassette recorders. I've never understood it -or perhaps I've made a bad comparison?

*****

The New South Wales Government has opted to use five different micro-computers in schools and other Government Departments. Happily, the Atari is one of the five for which contracts have already been let and will be in force until September 1985. The other four are the IBM, Apple, MicroBee and BBC computers. An effect of this contract can be seen in an excerpt from a letter from Peter Tucker of Lismore:

"...Our school has ordered 12 800XLs and goodies to replace our MicroBees. We will retain our two Apple IIs. Five staff members have also ordered 800XLs and a local private school has ordered 18 800XLs...".

Coupled with this, the availability of the new Atari 600XL through shops such as K mart and RetraVision is sure to increase support for the machine, especially in terms of software availability. I guess that we should also expect our club membership to increase from the 210 members it already has. To top all of this off FUTURETRONICS has announced (through managing directors Harry Chonja and Peter Alpar) "Our objective is to capture the major share of the Australian home computer market. We now have the products to do this and we know that the Atari will be a force to be reckoned with ... It is a commitment by ATARI and FUTURETRONICS Australia Pty Ltd to deliver to the marketplace the products that are needed."

*****

I've now heard from two seperate sources that FUTURETRONICS did not intend to release the new 1050 disk drive until existing supplies of the "old" 810 were sold up!

Bernhard Kirschner of COMPUTERWAVE advised me that he purchased the remaining stocks of 810s so that the 1050 would be released. He has since sold his stocks of 810s, but still has the 1050 for $699. An important thing to note is that, to date, DOS III has not been released in the States and may be an extra cost to owners of the 1050.

While on the subject of disk drives, I heard rumour that IMAGINEERING may start importing the double density RANA disk drives. A phone call to IMAGINEERING got some very "tongue in cheek" replies, the only assurance we have is that when and if they do import them, we'll all hear about it.

I've heard of a few people in other states importing the RANA and while it is very tempting to buy one, I'm not prepared to get one when I'm uncertain I'll be able to get it repaired in Australia. Some other impressive sounding double density drives released in the States are the TRAK, INDUS GT and PERCOM. Although some of them have had teething troubles, I'm wondering if they will make ATARI's 1050 obsolete -I mean why buy a drive which isn't a true double density drive when you can have true double density?

*****

As everyone already knows, ATARI has been losing vast amounts of money in the States, mainly through the 2600 video games machine, but also partly because they weren't able to get the new XL series onto the market quickly enough. The 800XL was released over there just prior to Christmas and is said to be available here already. I've heard from several sources that the XL series won't run all available software and that the new 1050 disk drive won't boot some software!

Rumor has it that ATARI's 1400 series has been scrapped, at least until ATARI is more economically stable. It's a pity since there were also some rumors of a new computer in the wind -a 16 bit business machine (the 1600???) which had compatibility with the present Atari computers, the IBM PC, the Apple II and CP/M. It was rumored to be released in time for Christmas 1983, and its other features were supposedly both 8088 and 6502 CPU chips, dual floppies, serial and parallel ports as well as RGB output(which give the same high quality displays as the arcade games).

*****

The former president of ATARI in the States is now the vice-president of Apple. Apparently he wanted to work for a company that has more of a commitment to computers!

*****

While ATARI licks its financial wounds, it has started to make use of owning the rights to programs such as Pac-Man, Defender, DigDug, Centipede and others. These games have been converted for the Vic 20, Commodore 64, Apple II, IBM and TI 99/4A and are marketed under the name ATARISOFT. FUTURETRONICS has also been advertising them over here.

*****

The RESTON computer group (the ones who marketed PAINT) are now marketing a program called MOVIEMAKER. Apparently you develop a "movie" frame by frame and can then play it back. One review I've read said it was very good, especially for computer novices. It's a two disk package and costs $US60.00 plus postage. For further details write to:

Marketing Communications
11480 Sunset Hills Rd
Reston, VA. 22090
U.S.A.

*****

Eugene ACE is selling a composite index to Atari related articles which have appeared in ANALOG, ANTIC, COMPUTE!, CREATIVE COMPUTING and the EUGENE ACE newsletters. SORT FINDER 1.2 is a printed article and not a program for the Atari. The index covers all the hardware and software articles which appeared in the above magazines between April 1981 and June 1983. Costs are $US7/copy or $US6 each for two or more -don't forget to allow some extra for postage! Write to:

Valley Soft
2660 SW DeArmond
Corvallis
OR. 97333
U.S.A.

*****

Just about everyone who has seen "The Book of ATARI SOFTWARE 1983" has been very impressed with the way it reviews just about every imaginable piece of software that's available for the Atari -especially its no-nonsense accuracy! Well, the 1984 book is now being advertised and will probably be available in Australia shortly for around $30. In the meantime it is available in the States for $US19.95 plus postage. Write to:

The Book Company
Dept. COM, 11223 S. Hindry Ave
Los Angeles
California 90045
U.S.A.

*****

It appears that that ANALOG magazine has gone monthly. I recently received a renewal form which gave costs of $US28/12 issues, $US52/24 issues and $US79/32 issues plus $US12 for surface postage. The Cassette version is $US90/year and the Disk version is $US130/year. A good magazine but it can take a while to get your subscription running. Write to:

ANALOG Computing
P.O. Box 23
Worcester,
MA. 01603
U.S.A.

*****

Compute! magazine is also good value at $US42/year (12 issues) and that's AIRMAIL!!!! Write to:

COMPUTE! Magazine
P.O. Box 914
Farmingdale, NY 11737
U.S.A.

*****

If you're after some paper for your printer at reasonable prices this place may interest you:

Illawarra Stationary Supplies Pty Ltd
356 Railway Parade,
Carlton
N.S.W.
Phone (02) 587 8255

Single ply, plain telex rolls are about $4.50 each. You can get a box of 2500 sheets of A4 60gsm (grams/square metre) computer paper for around $42.00.

*****

MODEMS

by Anthony Dinallo. (Haberfield, N.S.W.)

With the introduction of the Bulletin Board Service (BBS) now in its final stages, an understanding of the connection details and the relevant standards used throughout is necessary.

There is plenty of information available in the commercial computer magazines describing telecommunications concepts between computers, types of modems, software used, bulletin board services available and terminology explained. The May 1983 issue of ANTIC is a fine example, with over half the magazine devoted to communications.

What is often confusing though, are the types of standards used and whether a MODEM is compatible or not. I will give a brief description of the concepts and standards used and hopefully this will clear up the jargon when the club's bulletin board comes into service.

BASIC CONCEPTS

The telephone line provides us with a convenient medium with which we can transfer data from one computer to another. To take advantage of this readily available medium, all we need to do is connect the computer to the phone line. This is normally accomplished with the aid of a MODEM.

The word MODEM is coined from the words MOdulate/DEModulate and represents want happens to the data signal travelling to and from the computer.

The logic signals produced by the computer are too low in voltage to be able to be sent effectively over any distance greater than a few metres. Therefore, the signals are sent via the MODEM which modulates the data signals onto an audio carrier tone for transmission over the phone lines. At the receiving end, the MODEM demodulates the carrier and recovers the data signals.

THE CCITT V.21

In order to make data transmission over the phone lines compatible with various equipment, standards need to be set. The one generally used in Australia and for our purpose is known as CCITT V.21. (Consultive Committee in International Telegraphy and Telephony, V-series No.21). This standard simply specifies the rate of data transmission and the two sets of frequencies for transmitting and receiving. See table 1 below.

TABLE 1 - CCITT V.21

Transmission Rate: 300 bits per second.
Originate frequencies: LOGIC 1 980 Hz.
LOGIC 0 1180 Hz.
Answering frequencies: LOGIC 1 1850 Hz.
LOGIC 0 1650 Hz.

MODEMS that are set up for the CCITT V.21 are not compatible with the United States Bulletin Boards which use the BELL 103 standard. Therefore, enthusiasts wishing to communicate with overseas bulletin boards need to use so called 'WORLD' MODEMs which have selectable transmission formats. Such a MODEM was recently built by myself, and is described in the January issue of 'ELECTRONICS AUSTRALIA'.

The RS-232-C

Connecting the MODEM to the computer also needs to follow a standard, this comes in the form of the RS-232-C. (Recommended Standard of the Electronic Industries Association No. 232 rev. C). The RS-232 transfers data in serial form (one bit at a time), from one device to another, in this case from the the computer to the MODEM. Note that the ATARI serial port is NOT an RS-232-C standard!

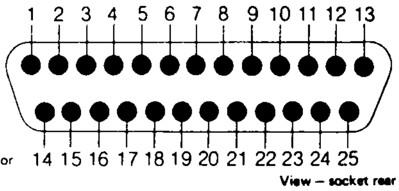
For serial data transmission only three wires are required, one for sending the data, one for receiving, and a common reference. Meanwhile, the RS-232-C standard specifies twenty two wires to be connected! Luckily for Personal Computer usage only eight are commonly used for communications purposes. Two of these wires are for data transmission and the other six are control wires. These wires are normally connected using DB-25 plugs and sockets. A DB-25 connector looks similar to the nine pin connector used for the ATARI joysticks (which is in fact a DB-9 connector), only it is wider - to accomodate twenty-five pins. The ATARI 850 interface module PORT 1 is RS-232-C compatible but it uses a DB-9 socket, therefore you need to make up your own cable to interface it to a DB-25 socket.

Table 2 shows the wires and connections used.

TABLE 2

RS232C CABLE PLUG CONNECTIONS

Pin No	Signal
1.	Protective Ground
2.	Transmit Data
3.	Received Data
4.	Request to Send
5.	Clear to Send
6.	Data Set Ready
7.	Signal Ground
8.	Received Line Signal Detector
20.	Data Terminal Ready



THE RS-232C INTERFACE

The basic RS-232C electrical specifications

Communication rate	0-20,000 bits per second
Driver output voltage levels, maximum no load.	−25V logic 1, +25V logic 0
Driver output voltage ranges for loads between 3k and 7k ohms.	logic 1: −15V (7k) and −5V (3k) logic 0: +15V (7k) and +5V (3k)
Driver output current, short circuit.	500mA maximum
Driver output impedance with power off.	300 ohms minimum
Maximum driver output slew rate.	30 volts per microsecond
Receiver input resistance	7k ohms maximum, 3k ohms minimum
Effective receiver input capacitance	2500pF maximum
Maximum receiver input voltage range.	−25V to +25V

Descriptions of the signals are as follows:

TRANSMIT DATA (TX): This is the wire that carries the actual data from the computer to the MODEM.

RECEIVED DATA (RX): This is the wire through which the computer receives the data from the MODEM.

REQUEST TO SEND (RTS): This is a signal from the computer to the MODEM indicating that the computer wishes to send data.

CLEAR TO SEND (CTS): The MODEM replies with a clear to send signal.

DATA TERMINAL READY (DTR): Indicates to the MODEM that the computer is in a state of readiness. (In other words it's ON!).

DATA SET READY (DSR): With this signal the MODEM indicates to the computer that it is also ready to send data over the communication Lines.

RECEIVED LINE SIGNAL DETECTOR (DCD): This signal (also known as Data Carrier Detect, or simply Carrier Detect) indicates that the MODEM has received the Audio Carrier Tone from the distant party, and has established a communications link.

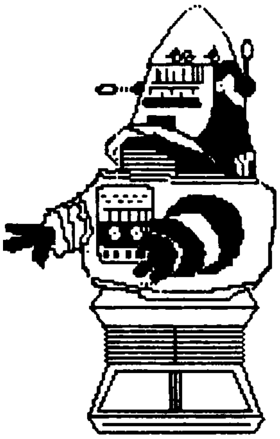
SIGNAL GROUND or COMMON GROUND: Provides the reference point for all the above signals.

The above signals are also referred to as 'handshaking signals' and are required so that data can be exchanged without any loss or confusion.

FURTHER INFORMATION

This guide is only intended as a brief summary, and further information on connecting to, and using a bulletin board service will be published when the club establishes its own BBS.

Meanwhile a study of past magazines such as ANTIC or COMPUTE will reveal the commonly used methods for using a BBS and the ATARI 850 INTERFACE MODULE USER MANUAL gives an excellent description on using the RS-232-C Interface. Readers who wish to experiment with their own hardware, or wanting to gain an in-depth understanding of data communications may find the book titled 'DATA COMMUNICATIONS FOR MICROCOMPUTERS' by Nichols, Nichols, and Musson, published by McGRAW HILL, an excellent reference guide.



BOOK REVIEW

ADVANCED 6502 INTERFACING

by John M. Holland
copy by PRENTICE-HALL(AUST.). $19.95.
reviewed by Anthony Dinallo. (Haberfield, N.S.W.)

The 6502 microprocessor is a versatile chip in the sense of hardware interfacing since hardware attached to the address bus can be addressed directly, as if it were a memory location.

The APPLE computer uses this capability with its vast range of plug in interface cards while the ATARI utilises this feature by addressing the GTIA, ANTIC, and POKEY chips directly.

John Holland has produced a book for the electronic enthusiast who wishes to experiment with external devices and peripherals which can be interfaced to the 6502.

Although the book doesn't mention the ATARI computer, the ATARI does use the 6502 and many of the applications will apply to it.

The first three chapters describe the system architecture of the 6502, and how it is used in microprocessor circuits. It explains such topics as Page Zero, The Stack, System Clock, Read/Write, Reset, synchronising, Interrupts, Input and Output design, Memory Addressing and many more. It shows connection techniques with circuit diagrams giving a pictorial description throughout the book.

The remaining five chapters go into depth to explain other topics such as SERIAL COMMUNICATIONS, TIMERS & TIMING, ANALOG TO DIGITAL CONVERSION, DATA ACQUISITION METHODS, DIGITAL TO ANALOG CONVERSION, and METHODS FOR DRIVING HIGH CURRENT LOADS, each with examples and diagrams showing common techniques used.

The appendices contain Data Sheets of the most popular interfacing integrated circuits, an ASCII Code Chart, and a summary of the 6502 Instruction Set.

Although this book may well be "over the heads" of many Atari users, it does provide an excellent description of microcomputer designs and techniques, and may prove valuable to the person who is very interested in learning and/or experimenting with hardware applications.

Interfacing Your Atari to the Real World

by Marshall Dubin. (Michigan A.C.E. Newsletter April-May, 1982)

BASIC INPUT

With the addition of a few parts and some external circuits, your Atari personal computer can send and receive signals from the "real world". These signals can be the monitoring of alarms, thermostats and a variety of digital and analog sources which in turn can control various motors, appliances, lights, etc. The purpose of this article is to discuss a method in which we can monitor up to four input signals from external sources in the real world.

For our purposes we will be using the joystick ports which are located on the front panel of the computer. Figure one shows a pin connection diagram of a single port. All four ports share this connection pattern.

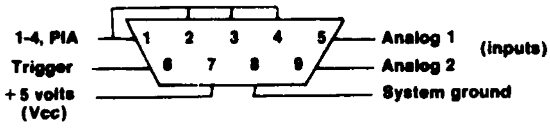


Figure 1: Pin Connections

Note that there are four pins labelled PIA. These pins are connected directly to the internal PERIPHERAL INTERFACE ADAPTER CHIP of the computer. The pins may be formatted for either input or output. We shall discuss them in more detail later. Additional pins are: the trigger pin for the joystick trigger buttons, two analog input pins, used for the paddle controllers, a +5v pin and a system ground pin. In this example we shall use the TRIGGER pins. Because there are four ports, each port may control a different device connected to the trigger pin of that port.

BASIC INPUT USING THE TRIGGER PIN

When the computer is first switched on, the trigger pins are at a logic 1 (high) state. This is the DEFAULT status of these pins. The normal procedure when using joysticks is to have the computer react if the pin reads LOW (logic 0). When this is the case, the trigger is assumed to have been pressed. The following listing illustrates this point:

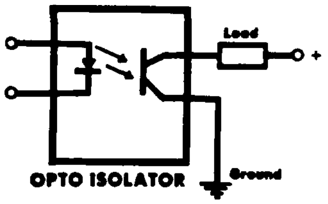
```
10 X=STRIG(0):REM for port one
20 IF X=0 THEN PRINT "TRIGGER IS ACTIVE"
30 PRINT "TRIGGER IS NOT ACTIVE"
40 GOTO 20
```

The computer will take one action when the trigger is pressed and do something else when it is not. Any game using joysticks illustrates this point. Now, what if an alarm sensor, liquid level sensor or light activated switch were connected to the joystick trigger input? Ah-ha. We begin to see that the old joystick port can do more than blow away Zylons.

THE BASIC INPUT CONNECTION

As I mentioned earlier, the default status of the trigger pin is a logic one. To simulate pressing the trigger button, we must make the status of that pin a logic zero. This is done by connecting that pin to system ground. When we do that, the pin is said to have been "pulled low", and will now read a zero.

The easiest way to do this is by connecting pin 6 of the controller port to pin 8 of the same port. In real world situations though, especially if the switch is not located right next to the computer, and especially if you are monitoring something more than a Transistor-Transistor Logic (TTL) level signal, this is best accomplished through the use of an external relay or better still, an OPTO-ISOLATOR. Let's look at opto-isolators first.



An opto-isolator is composed of a Light Emitting Diode (LED) and a phototransistor. When voltage is applied to the diode, it glows. The light emitted from the diode reacts with the photo-sensitive transistor and "biases" it into conduction. Here the transistor acts as a switch which is turned on by the light from the diode, and off when the diode is dark. Since power need only be given to the diode, there is no chance that a higher, damaging voltage can cross the optical barrier and damage the computer. As illustrated in figure 2, when the transistor switch is ON, pin 6 of the controller port is connected to pin 8 (ground) thereby pulling it low and simulating a trigger press.

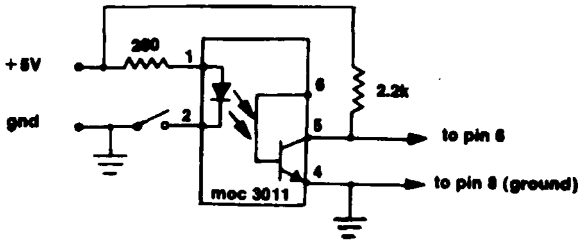


FIGURE 2: Using an opto-isolator
On this example, a relay is powered on by a high voltage such as 110 volts, and this causes the contacts to pull pin 6 low.

Do not try to power the opto-isolator from the "on-board" 5 volt power supply. It can only withstand 40 or so milliamps, and has other uses. Also note that the diodes in some opto-isolators can draw as much as 100 milliamps current, so you will need a supply capable of driving at least one or more of them. In addition, you may have to use a current limiting resistor between the opto-isolator supply and the LED, especially if you use higher voltages to drive the isolator. Use Ohm's law to figure out the resistance you will need for the voltage and current ratings you will be using, if they are different from mine.

Figure three shows a method of using a relay to pull the trigger pin low. Figure four shows a light activated sensor (Sargent and Shoemaker, 1981) which will also work well.

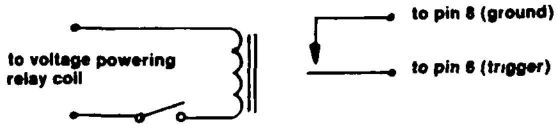


Figure 3: Use of a relay

When switch is closed, a signal is passed to the computer. Switch can be part of a relay if you wish to monitor high current devices. Have your device trigger a relay, which in turn will turn on the opto-isolator.

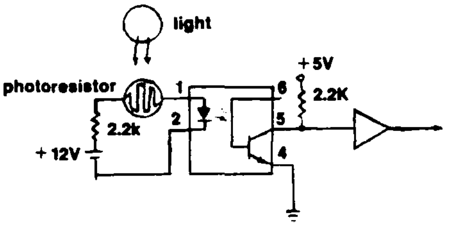


FIGURE 4

The on/off status of a light can be monitored by using a photoresistor as the LED's current limiting resistor. If the photo-resistor circuit uses more than five volts, an extra current limiting resistor may be needed (LED current shouldn't exceed 30 ma).

PROGRAMMING CONSIDERATIONS

Basically, the sensors can be used in a program in pretty much the same way as the joystick trigger buttons. The keyword "STRIG" will read the status of this pin as it does for the normal use of the joystick. Please note that the "PTRIG" keyword does not access pin 6 of the controller port, but uses different pins. You must use the "STRIG" keyword, or else "PEEK" locations 644, 645 or 647 (decimal) to read the pin status. Each location is for a different port. 644 is port one, etc. Much the same as "STRIG(0)". So now what?

Go to it! Your computer can read and react to all kinds of neat things beside Zylons or Space Invaders. Try using a light beam sensor as a counter, or determining how many times your furnace motor kicks on during the day, or reacting to a metallic "end of tape" sensor for programmable slide shows, or even bio-feedback (be SURE to use optical isolation!). What it boils down to is your own imagination. Experiment! Learn! Enjoy!

CONTROLLING POWER

We will now look at ways to use the joystick ports for output. This gives you a means to control a wide variety of external devices, relays and the like.

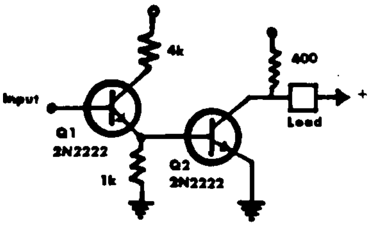
Referring back to figure one, you can see that the joystick port has several potential input sources available. For example, two of the pins are intended for use with the paddle controllers. These are called the ANALOG pins. They take an analog source such as a variable resistance and convert it into a digital signal. This is, in essence, how the paddles function. They provide a resistance via a potentiometer within the paddle unit, between the analog input pins and +5 volts DC. The computer interprets the variable voltage as a digital number between 0 and 228.

This is called "on-board" analog to digital conversion. Units performing a similar function may be purchased at a hefty price, but Atari owners have the use of 8 of these units built right in!

For now, let's concentrate on pins 1-4 on the joystick ports. These are the pins of the Peripheral Interface Adaptor Chip, more commonly referred to as the PIA. Basically, the PIA provides a means of connecting your computer to peripherals. The PIA chip can be programmed for either input or output. There are two PIA ports of eight bits each available for your use. Joystick ports 1 and 2 compose PIA port A, whilst joystick ports 3 and 4 compose PIA port B. Each port is one byte (8 bits) and may be used together or individually to provide input and output functions. Some of these functions may be used to drive a printer or other accessory, or even a series of power relays which can control alarms, lights, appliances, motors or whatever.

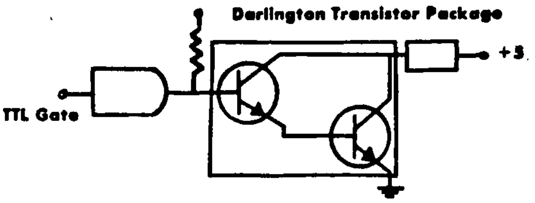
The problem involved in controlling larger interface devices is basically a problem of taking a small amount of power and amplifying it. The ports on your computer are not made to power anything more than another chip. The manual recommends a maximum of 1 TTL load (1.6ma.) for each PIA bit, and no more than 50ma. at the +5V pin. If we are using relays for controlling power, we will need at least 12-24 volts.

There are several ways to accomplish this task. The figures below illustrate some of them. The most common arrangement is the transistor driver.



OPEN COLLECTOR TRANSISTOR DRIVER

In this arrangement the computer provides a very small voltage which turns on the transistors, which in turn switch the load. A second way is through use of opto-isolators. The computer provides 5 volts which switches the LED of the isolator on. When the diode is lit, this triggers a photo-sensitive transistor which is connected to a larger load or relay. A DARLINGTON transistor package can be used in a similar fashion. The low voltage fires the Darlington transistor which can then switch a considerably larger current.

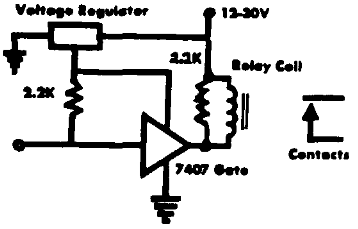


DARLINGTON DRIVER

Finally, there are integrated circuit interface chips such as the 7407 which allows a switching of up to 30 volts from the 5 volt TTL level of the Atari.

All of these methods will work and actually, which one to use will depend on the specific environment or task lined up for them. The method I prefer is the 7407 chip. It is inexpensive, will handle up to 30 volts and has six gates so that I may switch six devices from a single chip and will handle about 30ma, which will drive most small relays or solid state switches.

The SN7407 is an open collector device. To use it properly you must connect a 2.2K ohm resistor from each output to +5 volts. When an output is "on" the output is actually open - so the resistor supplies +5 volts to the device you are driving.



7407 DRIVER WITH RELAY

You can drive up to 30 volts at the outputs (but you may have to tamper with the value of the resistor somewhat). When an output is "off", it is shorted to ground and your device sees 0 volts (ground actually). The resistor limits this current to a fairly low value so you don't blow the power supply, or worse, the chip! Since the resistor can't supply much current, make sure the resistor/7407 combination is seen as the "ground side" of your circuit i.e. to drive a relay, connect +5V to one side of the relay and the other side to the output of the 7407. Then, to turn the relay on, turn the 7407 OFF. Current will still flow through the relay and then to the 7407 to ground.

You can easily drive discreet LED's with this too (such as for test lights), as well as a variety of small relays or solid state switches. Just make sure you SINK the current i.e. one end of your driven device goes to +5 (through a resistor!) and the other end to the 7407. Sending a "0" (logic level low) to the PIA turns the device ON, and a "1" (logic level high) turns it OFF. If you want to it the other way around, use the inverting 7406, which will turn your device ON with a high logic level and OFF with a low level. Recognize though, that the default state of the PIA when the computer is powered up is all bits high. If you are using an inverting 7406, your devices would come alive when you "powered up" the Atari. This is why I prefer to use the 7407 since I can power up and then have my software drive the devices by writing a "0" to the bit I want to power a device from.

Speaking of bits (and before you run off to warm up your soldering irons), a few words are in order concerning the structure of the ports. The PIA, as I mentioned earlier, consists of two ports; port A and port B (or PORTA and PORTB for all you mnemonics freaks). These are controlled through the use of the control registers for each port, PACTL and PBCTL. You may have heard of the PACTL because that's the one you poke with 52 to turn on the cassette player. The addresses are as follows:

PORTA 54016/$D300 - port A address
PORTB 54017/$D301 - port B address
PACTL 54018/$D302 - port A control
PBCTL 54018/$D303 - port B control

NOTE: a "$" in front of a number indicates that the number is in hexadecimal (HEX).

On power up the ports are initialized to $FFFF or all bits high. To use a port for input, just pull the bit of your choice low by connecting it to ground. To use the port for output it must first be formatted for output. This procedure is not complex:

1. POKE the control register (PACTL or PBCTL) with 56/$38.
2. Now POKE the port (PORTA or PORTB) with 255/$FF. This specifies that the port will be used for output.
3. POKE PACTL or PBCTL with 60/$3C.
4. Now, just POKE the port (PORTA or PORTB) with your data.

Essentially, you have a total of 16 bits to play with. Just remember that two joystick ports make up one PIA port. Sticks 0 and 1 are the A side whilst sticks 3 and 4 are side B. Each joystick port is 4 bits or 1 nybble. Each side of the PIA is 8 bits or 1 byte. When programming for output, you must remember that a specific BIT is a driving device. Therefore, one joystick port can drive 4 devices (1 for each bit). An entire PIA side will handle 8 devices if you use both A and B side you can trigger 16 individual devices at once or in any combination. You must POKE a decimal number whose BINARY representation will switch on a certain bit, or series of bits, into that port. For example, if I POKEd 255 into port A, all bits would be on. If I POKEd 12 into port A, only bits 3 and 4 would be on.

shadow registers, as follows:

Jack 1 (STICK 0) 632/$278
Jack 2 (STICK 1) 633/$279
Jack 3 (STICK 2) 634/$27A
Jack 4 (STICK 3) 635/$27B

You can also use the BASIC keyword "STICK" to access these ports eg. X=STICK(0), etc.

The program listing will provide you with a demonstration on how the ports are programmed. The program first allows you to select a port and program it for either input or output. Then you can write data to the selected port and the computer will peek the port and verify the data you wrote. Granted - this isn't elaborate, but it works!

REFERENCES

Sargent, M and Shoemaker, R - "Interfacing Microcomputers to The Real World", Addison-Wesley Publishing Company, Reading, Massachusetts, 1981.

Mosaic Electronics "Atari I/O Package", Mosaic Electronics, P.O. Box 748, Oregon City, Oregon. 97045.

"Programming and Interfacing the 6502", Marvin L. DeJong.

"Atari Operating System Users' Manual."

"Atari Hardware Manual."

EDITOR'S NOTES

In a future issue, Tony Dinallo intends to present an article on various hardware projects which will follow on from this article.

Michigan A.C.E. is a large users' group in the United States. Their address is:

Michigan A.C.E.
P.O. Box 2785
Southfield, Michigan 48037.
U.S.A

```
1 REM ################################
2 REM #            PIA4MAT            #
3 REM #       by Marshall Dubin       #
4 REM # Michigan A.C.E. Newsletter    #
5 REM #         April/May 1982        #
6 REM # Reprinted by A.C.E.(N.S.W.) #
7 REM #          February 1984        #
8 REM ################################
10 REM * PROGRAM TO FORMAT PIA PORTS
20 REM
30 GRAPHICS 0:POSITION 10,2
40 DIM IO$(10),DATA$(3)
50 ? "PIA PORT DEMO"
60 REM
70 REM PORT ADDRESS
80 REM
90 PORTA=54016:PORTB=54017
100 REM
110 REM * ROUTINE TO CONFIGURE PORT
120 REM
130 TRAP 130:? :? "Configure which por
t (1-4)";
140 INPUT PORT:IF PORT<1 OR PORT>4 THE
N 130
150 REM
160 REM SELECT PORT CONTROL REGISTER
170 REM ADDRESS (PACTL,PBCTL)
180 REM
190 IF PORT<3 THEN PCTL=54018:PORT=POR
TA
200 IF PORT>2 THEN PCTL=54019:PORT=POR
TB
210 ? :?
220 REM
230 REM SELECT INPUT OR OUTPUT
240 REM
250 REM ?"Input or Output";
260 TRAP 250:INPUT IO$
270 IF IO$(1,1)="I" THEN F=0:GOTO 340
280 IF IO$(1,1)="O" THEN F=255:GOTO 34
0
290 GOTO 250
300 ?
310 REM
320 REM CONFIGURE THE PORT
330 REM
340 POKE PCTL,56
350 POKE PORT,F
360 POKE PCTL,60
370 ? :?
380 REM
390 REM ENTER YOUR DATA
400 REM
410 IF IO$(1,1)="I" THEN ? "PORT IS FO
RMATTED FOR INPUT":? :GOTO 130
420 ? "NOW ENTER YOUR DATA"
430 ? "(ENTER A RETURN TO DO ANOTHER P
ORT)"
440 INPUT DATA$:IF DATA$="" THEN ? CHR
$(125):GOTO 130
450 TRAP 530
460 REM
470 REM POKE DATA TO PORT/VERIFY IT
480 REM
490 POKE PORT,VAL(DATA$)
500 ? "VERIFY ";PEEK(PORT)
510 GOTO 440
520 END
530 TRAP 40000:? "INPUT ERROR, RE-ENTE
R ";:GOTO 440
```

FOR SALE

*****

SoftSide Adventures: These are text only with some colour and sound effects. They are not easy! All are originals on 32k cassette for $7.50 each. The Adventures available are: James Brand Adventure, Dateline, Titanic, Arrow One, The Mouse That Ate Chicago, Alaskan Adventure and Danger Is My Business.

Tricky Tutorials: (Santa Cruz Educational Software) - 16k cassette $15 each. Tutorials available are: #1 Display Lists, #2 Horizontal and Vertical Scrolling, #3 Page Flipping, #4 Basics of Animation, #6 Sound.

OTHER: Rear Guard (Adventure International) - 16k cassette $15, Asteroid Miners (MMG Micro Software) - Mediocre arcade game with good 40 page tutorial. 32k disk $25. For more details, see me at a meeting or ring (02) 789-1397 (A.H.).

*****

PRINTER: Atari 822 thermal printer. 6 months old $150.

ATARI PILOT: Programming language $50.

DISK BASED GAMES: $20 each - Bug Attack, Jawbreaker, Apple Panic, Pathfinder and Raster Blaster.

CARTRIDGE BASED GAMES: $25 each - Centipede, Pac Man and Galaxian.

Phone Ingo Oestreich on (02) 78 5217 for further details.

*****

A Program Line Editor

by Chris Fitzgerald. (Clyde, N.S.W.)

Isn't it annoying when you type in a program listing from a magazine only to find that it crashes when you try to run it? Usually there is nothing wrong with the original program, you have just make a few typing errors; this is especially the case where you've had to type heaps of DATA statements.

I got sick and tired of going through programs looking for my typos and decided that my ATARI could help me out. The result of this was a program I called LINEDITA. The aim of the program was to make it easy to see the information you want to check and to reduce eye fatigue. It uses around 4.5k of memory and will be useable on about 90% of the programs you type in, provided it is numbered below 32000.

DIRECTIONS

After you type in LINEDITA, LIST it to cassette or disk. Later, when you have a program (with typos) in memory you can append LINEDITA using ENTER "C:" or ENTER "D:LINEDITA.LST" as applicable. You can then check your program by typing GOTO 32000 (followed by [RETURN] of course).

THE MAIN MENU

The first thing screen is the main =menu which presents you with 5 options.

Option 1 allows you to step through your program one line at a time. It presents a green screen which lists only one program line.

Option 2 allows you to step through the elements of data in the DATA statements. It presents a black and orange screen. Each element of data (to a maximum length of 20 characters) is displayed in GRAPHICS 2 for ease of reading. Since GRAPHICS 2 converts ATASCII control characters to ASCII characters (as well as reading inverse, normal, upper and lower case characters as being in different colour registers) I included a line of GRAPHICS 0 as an added check. At the bottom of the screen the entire program line containing the DATA statement is listed with a moving "cursor" to high-light the current data element. This does not act as a true cursor though, since it only converts normal characters (with ASCII value less than 128) to inverse, and not vice versa. I should also mention that if the first statement on the line isn't a DATA statement, the first part of the line will be overprinted by the "cusor" - don't worry, this won't change the program line in memory.

With both these options you'll first be prompted to enter a starting line number. If the line number you enter does not exist, the program will automatically go to the next line number (provided there is one!). Once the program searches out the desired line, the appropriate screen will be displayed.

Also, both these screens list 3 options at the top of the screen, they are:

Press "ESC" at any time (except during EDIT mode) to return to the main menu.

Press "E" to edit the current line. This will give you full control of the keyboard and away you go. When you're done, just keep hitting RETURN until the cursor passes over the "CONT" statement and you'll get back into the program where you left off. There are NO OPTIONS available in this mode. If you have been checking DATA elements, the EDIT screen will also display the DATA OFFSET within the current line and the DATA ELEMENT at which you stopped.

If extensive editing is carried out in this mode, the program may fail to list the line after acting on the "CONT" statement, if this happens simply return to the MAIN MENU and begin again from that line number.

Press the "SPACE BAR" to step through the program lines or data elements (according to the option you selected).

Option 3 allows you to list the resident program to cassette.

Option 4 allows you to list the resident program to disk after entering an appropriate filename.

On being presented with the main menu, the first thing you should do is LIST your program to cassette or disk using one of these options. Doing this before you start will ensure that you still have a copy of your program if something should go wrong. It will also clean out your variable table and reduce the length of the program to its minimum - something the SAVE command doesn't do!

Option 5 permits you to exit the program, clearing all the variables as well as giving you the standard GRAPHICS 0 screen. You will NOT be able to exit the program if at any time you entered the EDIT mode and didn't LIST the program to disk or cassette afterwards.

PEEK LOCATIONS USED

The following PEEK locations may be of interest to the curious:

136 and 137: the low and high bytes for the address of the first line number of the resident program.

182: the offset to the current data element in the current DATA line.

183 and 184: the low and high bytes which give the line number from which data is currently being read.

195: gives the number of the error which occurred or which triggered a TRAP statement.

560 and 561: the low and high bytes for the address of the start of the screen display list.

703: POKE with 24 for a full GRAPHICS 0 screen. POKE with 4 to force a text window at the bottom of the screen - all other numbers are ignored in this location. To print to the area above the "text window" you must use "? #6;", just as in GRAPHICS 1 or 2.

709: sets the brightness of the printing in GRAPHICS 0. In GRAPHICS 1 or 2 it determines the colour to be used when printing lowercase characters (i.e. SETCOLOR 1).

710: sets the background colour in GRAPHICS 0. In GRAPHICS 1 or 2, it determines the colour to be used when printing inverse upper case characters as well as the background colour of the text window (i.e. SETCOLOR 2).

712: in GRAPHICS 0, sets the border colour. In GRAPHICS 1 or 2 it sets the background colour (i.e. SETCOLOR 4).

752: makes the cursor visible when POKEd with a 0 and makes it invisible when POKEd with a 1 - this takes effect after the next PRINT (?) command.

764: holds the internal code of the last key pressed i.e. 255 indicates that no key was pressed while 33 indicates the SPACE BAR. Note that all the keys have their own internal code.

ALTERING LINEDITA TO CHECK HIGH LINE NUMBERS

In order to use LINEDITA to check a program with line numbers above 75, there are several changes needed. You'll need a line renumbering program which automatically updates GOSUBs, GOTOs, etc.

1. Change "LINEDITA.LST" to "LINEDITB.LST" in line 32001.

2. Renumber the club header in lines 32000-32006 down to line numbers 0-6. Then list lines 0-6 to disk using 'LIST "D:filename.ext",0,6' or to tape using 'LIST "C:",0,6'.

3. Delete lines 0-32006 from memory to leave lines numbered 32010 and up in memory. Then, alter the following lines:

Line 32020: Change LINEDITA to LINEDITB.
Line 32130: Change B=0 to B=-73.
Line 32150: Add TRAP 30 to the start of the line.
Line 32160: Delete this line.
Line 32420: Change D<0 to D<76
and D>=32000 to D>32767.
Line 32490: Change LIST C$,0,31999
to LIST C$,76,32767.
Line 32510: Change LIST "C:",0,31999
to LIST "C:",76,32767.

4. Renumber the resultant program to start at line 10 in increments of 1. Line 24 should generate a "NO CORRESPONDING LINE NUMBER" message (for the TRAP 30 you put in) while lines 31 and 42 should generate an "INDIRECT LINE REFERENCE" message (for the RESTORE B commands).

```
32000 REM ############################
32001 REM #      LINEDITA.LST        #
32002 REM #  by Chris Fitzgerald     #
32003 REM #   Published by Atari     #
32004 REM #  Computer Enthusiasts    #
32005 REM # (N.S.W.) February 1984   #
32006 REM ############################
32010 CLR :DIM A$(20),B$(20),C$(14),D$
(20),E$(20):U=710:V=752:W=712:X=125:Y=
764:Z=255
32020 GRAPHICS 0:POKE V,1:POKE U,224:P
OSITION 11,1:? "LINEDITA UTILITY":POSI
TION 13,3:? " MAIN MENU ":? :?
32030 ? "1 CHECK PROGRAM LINE BY LINE"
:? :? "2 CHECK DATA STATEMENTS ONLY":?
 :? "3 LIST PROGRAM TO CASSETTE":?
32040 ? "4 LIST PROGRAM TO DISK":? :?
"5 TO EXIT PROGRAM":? :? "ESC TO RETUR
N TO MENU":POKE 703,4
32050 POP :POKE W,0:POKE V,1:? CHR$(X)
;"MAKE YOUR SELECTION":POKE Y,Z
32060 A=PEEK(Y):IF A=Z THEN 32060
32070 ? CHR$(X):IF A=31 THEN C$=" LINE
 CHECK ":GOTO 32130
32080 IF A=30 THEN C$=" DATA CHECK ":G
OTO 32230
32090 IF A=26 THEN 32500
32100 IF A=24 THEN 32480
32110 IF A=29 THEN ? " EXIT PROGRAM ":
GOTO 32650
32120 GOTO 32050
32130 GOSUB 32410:POKE 703,24:? CHR$(X
):B=0:C=PEEK(136)+256*PEEK(137)
32140 POKE V,1:POKE U,192:GOSUB 32470:
POSITION 0,8:? "      Number of lines 
checked         "
32150 LINE=PEEK(C)+256*PEEK(C+1):B=B+1
:IF LINE<D THEN C=C+PEEK(C+2):GOTO 321
50
32160 IF LINE>=32000 THEN 32220
32170 POSITION 36,8:? B:POSITION 0,14:
? CHR$(156);CHR$(156):POSITION 0,13:LI
ST LINE:POKE Y,Z
32180 A=PEEK(Y):IF A=42 THEN ? CHR$(X)
;C$:FLAG=1:GOTO 32440
32190 IF A=33 THEN C=C+PEEK(C+2):GOTO 
32150
32200 IF A=28 THEN 32020
32210 GOTO 32180
32220 POKE W,34:? CHR$(X):POSITION 11,
12:? " END OF LISTING ":GOSUB 32550:GO
TO 32020
32230 GOSUB 32410:POKE 703,24:? CHR$(X
);:LINE=D:RESTORE LINE:IF LINE=0 THEN 
RESTORE
32240 DL=PEEK(560)+256*PEEK(561):FOR L
=DL+10 TO DL+13:POKE L,7:NEXT L:POKE D
L+17,6:POKE DL+18,6
32250 TRAP 32390:POKE DL+25,65:POKE DL
+26,PEEK(560):POKE DL+27,PEEK(561):POK
E U,244:OLD=-1:GOSUB 32470
32260 B$="                    "
32270 READ A$:C=PEEK(182):LINE=PEEK(18
3)+256*PEEK(184):POKE Y,Z:POKE V,1
32280 IF OLD<>LINE THEN OLD=LINE:GOSUB
 32400:E=LEN(STR$(LINE))+8:F=12
32290 E$=A$:IF LEN(E$) THEN GOSUB 3237
0
32300 IF E$="" THEN D$=",":E=E-1
32310 IF E>39 THEN E=E-38:F=F+1
32320 POSITION E,F:? D$:POSITION 20,5:
? B$;:POSITION 20,5:? A$;:POSITION 2,8
:? B$:POSITION 2,8:? A$
32330 A=PEEK(Y):IF A=33 THEN POSITION 
E,F:? E$:E=E+LEN(D$)+1:GOTO 32270
32340 IF A=42 THEN GRAPHICS 0:POKE U,1
92:FLAG=2:RESTORE LINE:? C$:? :? " OFF
SET  ";C;"   DATA  ";A$:GOTO 32440
32350 IF A=28 THEN 32020
32360 GOTO 32330
32370 D$=E$:FOR L=1 TO LEN(E$):IF ASC(
E$(L,L))<128 THEN D$(L,L)=CHR$(ASC(E$(
L,L))+128)
32380 NEXT L:RETURN
32390 ? CHR$(X):POKE U,34:POSITION 2,8
:? "ALL DATA STATEMENTS HAVE BEEN CHEC
KED":GOSUB 32550:GOTO 32020
32400 ? B$;:? B$:POSITION 2,12:? CHR$(
156);CHR$(156):POSITION 2,11:LIST LINE
:RETURN
32410 TRAP 32410:CLOSE #1:POKE V,0:? C
HR$(X);C$:? "START FROM WHICH LINE ";:
G=5:GOSUB 32560:D=VAL(B$):D=INT(D)
32420 IF D<0 OR D>=32000 THEN 32410
32430 ? CHR$(X):RETURN
32440 POSITION 2,11:LIST LINE:? :? :?
"CONT":POSITION 2,7:? "YOU MAY EDIT TH
E FOLLOWING LINE WHEN"
32450 POKE Y,Z:POSITION 0,7:POKE V,0:E
ND
32460 B=B-1:? CHR$(X);:ON FLAG GOTO 32
140,32240
32470 ? C$;" OPTIONS":? :? "ESC FOR ME
NU":? "E TO EDIT CURRENT LINE":? "SPAC
E BAR TO STEP THROUGH PROGRAM":RETURN
32480 TRAP 32540:POKE W,192:B=1:POKE V
,0:? CHR$(X);"INSERT DISK AND ENTER FI
LENAME":? "THEN RETURN":?
32490 G=12:GOSUB 32560:? CHR$(X);" LIS
TING TO DISK ":C$="D:":C$(3)=B$:POKE W
,14:LIST C$,0,31999:GOTO 32050
32500 TRAP 32540:POKE W,192:B=2:? "LOA
D CASSETTE - SET FOR RECORD":? "THEN P
RESS RETURN":POKE Y,Z
```

```
32510 IF PEEK(Y)=12 THEN ? CHR$(X);" L
ISTING TO CASSETTE ":POKE W,14:LIST "C
:",0,31999:GOTO 32050
32520 IF PEEK(Y)=28 THEN 32050
32530 GOTO 32510
32540 CLOSE #1:POKE W,34:POKE V,1:? CH
R$(X);"ERROR # ";PEEK(195):GOSUB 32550
:? CHR$(X):ON B GOTO 32480,32500
32550 FOR L=1 TO 500:NEXT L:RETURN
32560 OPEN #1,4,0,"K:":B$="":L=0
32570 POKE Y,Z:GOSUB 32620:IF A=27 THE
N CLOSE #1:GOTO 32050
32580 IF L<G THEN IF A=46 OR A>47 AND
A<58 OR A>64 AND A<91 THEN L=L+1:? CHR
$(A);:B$(L)=CHR$(A):GOTO 32570
32590 IF L>0 AND A=126 THEN ? CHR$(30)
;" ";CHR$(30);:B$(L)="":L=L-1:GOTO 325
70
32600 IF A<>155 THEN 32570
32610 CLOSE #1:RETURN
32620 A=PEEK(Y):IF A=39 OR A=Z THEN 32
620
32630 IF A>127 THEN A=A-128
32640 GET #1,A:POKE 767,0:RETURN
32650 POKE W,34:IF FLAG THEN ? "LIST P
ROGRAM TO DISK or CASSETTE FIRST":GOSU
B 32550:GOTO 32050
32660 POKE V,0:? "ARE YOU SURE? (Y/N)
";:G=1:GOSUB 32560:IF B$<>"Y" THEN 320
50
32670 CLR :GRAPHICS 0:END
```

Hardware Modifications

by Kenji Baba.
32 Koolpinyah Crescent, Tiwi, Northern Territory, 5792.

[Here's a potpourri of interesting hardware modifications which were described in two letters I received recently from Kenji Baba. If you are interested in the modifications, but are not proficient with a soldering iron, Kenji will make them for you. The prices are $40 for the double cartridge adapter, $18 for the cartridge PC boards (without EPROMs) and $30 for the monitor utility. He is also working on an EPROM burner and has promised some news on this later. All enquiries should be sent direct to Kenji at the above address or ring (??)27-4604 - GF]

Double Cartridge Adapter for ATARI 400

Description:

The adapter is a double sided printed circuit board with two cartridge connectors, two isolating switches and LEDs with gold plated edge connector that will fit into the ATARI 400's cartridge slot. The board is designed so that it will activate the limit switch when in the slot and requires no other parts to keep power on.

Once in the slot, it will provide a flexible cartridge interface. As well as providing a right hand cartridge slot as for the 800, it allows either cartridge to be turned on and off at the user's whim. The isolating switches control power and some signal lines so that the cartridge can be completely turned off (e.g. while it is inserted). With this isolating capability, 400 owners can use the cartridges in a very flexible and interesting way which cannot be done on the 800.

Applications:

1. Turn ATARI 400 into 800 by providing left and right cartridge slots.
2. Power up the computer with cartridge(s) in the connectors with the isolating switch(es) off to avoid cold starting the cartridge.
3. 48k owners can test cartridge programs in RAM by manipulating the switches provided.
4. Swap cartridges at any time by using the isolating switches with the power on without fear of damaging the cartridge or the computer.

Printed Circuit Board for Do-It-Yourself Cartridges

This is a double sided board that will fit into the ATARI cartridge slots. The board is designed to take one or two 2732 EPROMs and is functionally identical to the ATARI cartridges.

Monitor Utility in Right Hand Cartridge

This cartridge is for the right hand cartridge slot. It uses the above board and contains one programmed 2732 EPROM with space for another for custom routines. It works with or without BASIC, Assembler, etc. in the left hand cartridge slot and has a hexadecimal calculator plus the following commands - Go, Move, Compare, Dump, Disassemble, Change, Fill and Search.

Changing Operating System ROMs to EPROMs

As all of us know, 400's and 800's do not use addresses $C000 to $CFFF (a total of 4k). The OS ROMs comprise two 4k ROMs and one 2k ROM, all of which are 24 pin devices which are almost pin compatible with 2732 and 2716 EPROMs.

I have changed all of the ROMs of my 400 to 2732 (4k) EPROMs for a total of 12k, thus creating an extra 2k that can hold any program. I have mapped this 2k into $C800 to $CFFF as it requires less modifications. Pins affected by the modifications are 18, 20 and 21 and basically provide an inverter for the line going to pin 18. My modifications consisted of cutting a few tracks and providing a half dozen jumpers and 74LS00 or 74LS04 to invert chip enable signals so that the EPROMs would work. As there was no space for the extra chip, I had to piggyback it on one of the other chips. (Modifications may be easier for the 800.) In the EPROMs, I have changed the vector at $F223 - the pointer to the blackboard routine - to $C800 and put my own 2k monitor program in $C800 to $CFFF. In theory, it is possible to use the other unused 2k, but physically there is no room until 24 pin 8k EPROMS become available. I suspect that all new XL series machines use this unused area for the help and test routines.

Now, whenever there is no cartridge inserted or I type BYE from BASIC or Assembler, the system jumps to my own monitor where I can do all the memory manipulations - change, dump, fill, move, search, disassemble, etc.

Tips on your Program Recorder

from FUTURETRONICS Australia Pty Ltd. (N.S.W. Service Department)

Editor's Note: Some months ago we were fortunate to have the presence of Peter and Karl, from FUTURETRONICS N.S.W. The following is a reprint of the notes they gave out at that meeting and is included for the benefit of members who weren't able to be present.

****

PROBLEMS

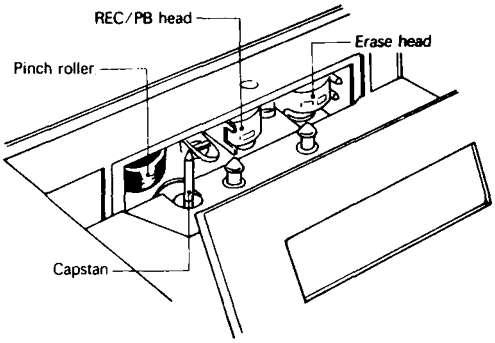
a) The recorders are prone to knocks and bumps. If the recorder is bumped, the data is altered during transmission because the head loses contact with the tape. Hence, you should keep the recorders away from vibrations.

b) The 410 is prone to errors 138 and 143 which are device timeout and checksum errors. These errors occur when the data being sent has an inncorrect baud rate.

These errors are the fault of the 410. The motor regulating is poor, therefore, the data being transmitted has variations. An effective modification is available at FUTURETRONICS to rectify this problem. This does not apply to the 1010.

MAINTENANCE

It is imperative that the tape recorder be cleaned at regular intervals. This includes the head, erase head, capstan and pinchroller.



Isoprophyl Alcohol is a good cleaner - Methylated Spirits is equally as good (economical).

Do not use cassette head cleaners. They often contain abrasives which rub the head and may destroy the quality of the heads.

For normal usage clean every month and for heavy usage clean every week.

Magnetic Fields: The recorders should be placed away from air conditioners, TVs etc. These radiate R.F. energy and alter the data transmission of the recorders.

GENERAL

Tapes should be rewound after use to prevent stray R.F. signals from making direct contact with the tape and to prevent dirt and grime making direct contact with the tape.

These would cause momentary dropouts and/or complete erasure.

CrO2 Tapes: CrO2 tapes are incompatible with the program recorders since they have a high bias frequency of 70 us. Use only Ferrite tapes which have a normal bias frequency of 120 us.

USER HINTS

1. System Reset

Problem: Pressing SYSTEM RESET does not reset the data I/O line in POKEY. Subsequent use of CSAVE is unreliable because that data I/O line is not clear, POKEY sends garbage, and the data stored is unrecoverable.

Solution: Avoid using SYSTEM RESET. Before using CSAVE or CLOAD, always execute an LPRINT command.

NOTE: Executing a serial bus command properly resets POKEY and clears the data I/O line. The simplest serial bus command to execute is LPRINT. If a printer is not attached when the LPRINT is executed, an 'ERROR 138' occurs. This occurrence is normal and does not interfere with the reset of POKEY.

2. Pre-recorded Tape Positioning

Problem: Tape is improperly positioned.

Solution: Pre-recorded tapes should load properly if rewound completely. If not, manually wind the leader onto the take-up reel before attempting the load.

CASSETTE FILES

A cassette file can be opened for reading or writing, but never both.

Data is held on the cassette tape in 128 character blocks. The last block of data or program file is the End of File (EOF) block and this is generated automatically by the computer. When reading a cassette, each block is read into the cassette buffer. This buffer is in memory from location 1021 to 1151. Data to be stored on the cassette tape is written from the cassette buffer. It follows that data to be stored must first be placed in the buffer using PRINT #CH or PUT #CH. Similarly, data to be loaded (read) into memory must access the buffer using INPUT #CH or GET #CH.

THE OPEN INSTRUCTION.

Format: OPEN #channel,task,aux,device

Examples: OPEN #1,4,0,"C:"
O.#CH,8,128,"C:"

There are eight channels available, numbered 0 to 7.
BASIC reserves channel 0 for the editor (device E:)
6 for screen graphics
7 for program loading/saving/
some printer operations.

(A table of these commands is given over page -ed.)

DEVICE	TASK	AUX
Program Recorder (C:)	4 Read 8 Write	0 Normal inter-record gaps 128 Short inter-record gaps
Disk file (D(n):filename(.ext)	4 Read 6 Read directory 8 Write-new file 9 Write-append 12 Read and Write	0 Ignored
Screen Editor (E:)	8 Screen output 12 K:input,S:out 13 Screen I/O	0 Ignored
Keyboard (K:)	4 Read	0 Ignored
Printer (P:)	8 Write	0 Normal characters 83 Sideways (ATARI 820)
RS-232 (R(n):)	5 Concurrent read 8 Block write 9 Concurrent write 13 Concurrent I/O	0 Ignored
Screen display (S:)	8 Normal 12 Read from screen 24 Text window 28 Window plus read 40 As above 44 but screen 56 is not 60 cleared	0 BASIC graphics 0 1 BASIC graphics 1 2 BASIC graphics 2 3 BASIC graphics 3 4 BASIC graphics 4 5 BASIC graphics 5 6 BASIC graphics 6 7 BASIC graphics 7 8 BASIC graphics 8

Note 1. Graphics 0 has no seperate text window.
2. The screen is always cleared in graphics 0.

PROGRAM REVIEW

Problems with AtariWriter

Review by Chris Fitzgerald. (Clyde, N.S.W.)

Just about every magazine you pick up lately is raving about how great ATARI's new word-processor, AtariWriter is. Just in case you hadn't realised it, this issue of INSIDE INFO has been written and formatted using AtariWriter and while I've been working with it, I've found some annoying little faults. I am not going to add to the praise AtariWriter has already received, you've more than likely heard about its good points anyway! Rather, here are some of it's problems.

The Printer Driver

When AtariWriter was first released, a few magazine reviews mentioned that ATARI had blundered, in as much as they had only allowed for ATARI's own printers. It appears that ATARI always intended to provide a disk of "printer drivers" to configure other makes of printers such as the EPSON, but it took a while before they became available. Hence, reviewers weren't able to assess its use fully.

The first problem is a very serious one - the EPSON printer driver (which I use) chews up almost 6.5 K! This leaves a little more than 14K for use, you may think that that is plenty to work with, but the "Interfacing the Atari to the Real World" article, in this issue, had to be split into two files because AtariWriter couldn't handle it.

The INFORMATION sheet (on the back of the front cover) is only one page in two columns, containing 650 words with the file being 4,900 bytes long. I was not able to use AtariWriter's "PRINT PREVIEW" option to preview the format of the whole page because there was not enough available memory! This makes this great little feature of AtariWriter useless. You can't get around it by using one of the in-built printer drivers to conserve memory since the formatting will be entirely different.

I can't help thinking that ATARI should have left out the built-in printer drivers and conserved AtariWriter's memory capacity by putting all of the drivers on disk!

Apart from this, there is the extra cost. Here in Australia, the cost of the printer driver disk is a phenomenal $A50.00 -half the price of Atariwriter itself!!! Even so, it's still cheaper than LETTER PERFECT which is listed as $300.00. In the States, AtariWriter sells for about $US65.00 with another $US25.00 for the printer driver, this totals $US90.00. I imagine that people there would rather buy LETTER PERFECT which they can buy for as little as $US67.00 -and this includes the printer driver!

Right Justification

Moving on, there is also a problem with RIGHT JUSTIFICATION. You will note that the Editorial page of this issue is printed using proportional font which is right justified, you may also notice that some of the spaces between words are absolutely HUGE and give the appearance of "white rivers" running through the print. That is one of the reasons why this issue of INSIDE INFO has been printed in condensed font -to make this problem less obvious.

Booting AtariWriter

If you are using a disk drive and you inadvertantly try to power up AtariWriter using a disk which doesn't have DOS on it, your drive will spin forever -the only indication of an error is that the blue border around the ATARI symbol, on the opening screen, turns purple.

Underlining

Then there is this other problem, you can't underline blank spaces, I'm sure you'll agree this looks terrible -especially in headings. For INSIDE INFO, I've had to load a line of "inverse spaces" from a LISTed program and then use AtariWriter's ability to relocate (or move) characters to the position where the "underlined space" should occur. I suppose I could always embed my printer control code into the text -making another feature of AtariWriter useless.

Formatting Commands

All underlined characters in AtariWriter appear on the screen as inverse video characters, unfortunately, so do most of AtariWriter's formatting commands -i.e. an inverse (control) "L" followed by "10" *indicates that your document's left margin* is set to the 10th print position.

These inverse formatting characters are put into the text by hitting the CTRL key along with the appropriate character. They have the nasty habit of making their way into a string of underlined characters and can't be seen on the screen. When you go to print the underlined characters you'll notice that some characters may go missing on your print-out or your printer will spit out a largely blank page -this being caused by an inverse "E" which AtariWriter sees as a "forced end of page". You might think that's just a dumb mistake I've made, but other people have had the same problem. A solution? -these control characters can be found. If you turn-off the underlining (CTRL U) for a string of characters, the control character will remain in inverse video on the screen.

Double Column Printing

One of AtariWriter's great features is that it can print two columns on the one page -there are two problems here.

1. The formatting control characters are always followed by RETURN which means that they take up a physical line. This extra line shows up with the first column of print being one line lower than the second column. If you leave out the RETURN after the formatting blocks you can't always control the font style (i.e. condensed font may end up as proportional font).
2. AtariWriter prints the two columns one at a time. The first column is printed to the end of the page then the printer must reverse line-feed the entire page before printing the second column. If your printer can't do reverse line feeds you can't use the two column option. If your printer can use this option, you must be careful when using sheet paper -it may catch on the printer's paper-end detector while reverse feeding. I have to print A4 sheets on foolscap to avoid this problem.

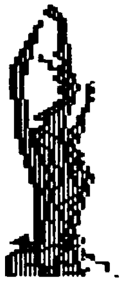
I can't understand why AtariWriter doesn't print the two columns simultaneously, from what I've heard ATARI's original word processor could.

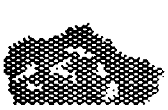
Turning the Page

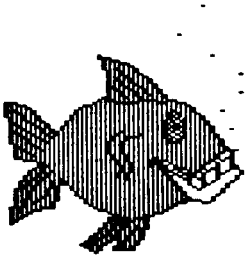
I've found that AtariWriter automatically turns off double-strike or emphasised modes when it begins a new page. The only way around this is to repeat the formatting block within your text files wherever a new page will start. Apart from this, I've found that the printer driver for the EPSON won't allow printing in double-strike "normal" font though it will allow empasised normal font.

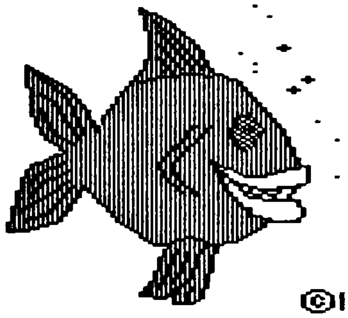
An End

I'm sorry if I've made you feel bad about AtariWriter, it's still a good word processor, and the only one that I've used, it's just a pity ATARI didn't make it a GREAT word processor. In Australia it kills the opposition for value, but I think it would be a different proposition in the States! In the meantime, I'd just love to see how LETTER PERFECT would work on my system -if only for a comparison!! To be fair, perhaps someone could write a review of Letter Perfect, especially with regard to similar faults, for the next issue.









©1983 JBB